BALANCED HEDGING AND TRADING BASKETS

David H. Bailey
Complex Systems Group Leader - Lawrence Berkeley National Laboratory
dhbailey@lbl.gov

Marcos M. López de Prado
Head of Global Quantitative Research - Tudor Investment Corp.
and
Research Affiliate - Lawrence Berkeley National Laboratory
lopezdeprado@lbl.gov

This version: July 2012

ABSTRACT

A basket is a set of instruments that are held together because its statistical profile delivers a desired goal, such as hedging or trading, which cannot be achieved through the individual constituents or even subsets of them. Multiple procedures have been proposed to compute hedging and trading baskets, among which balanced baskets have attracted significant attention in recent years. Unlike Principal Component Analysis (PCA) style methods, balanced baskets spread risk or exposure across their constituents without requiring a change of basis. Practitioners typically prefer balanced baskets because their output can be understood in intuitive terms.

We review three methodologies for determining balanced baskets, analyze the features of their respective solutions and provide Python code for their calculation. We also introduce a new method for reducing the dimension of a covariance matrix, called *Covariance Clustering*, which addresses the problem of numerical ill-conditioning without requiring a change of base.

1. INTRODUCTION

A basket is a set of instruments that are held together because its statistical profile delivers a desired goal, such as hedging a risk or trading it, which cannot be achieved through the individual constituents or even subsets of them. Multiple procedures have been proposed to compute hedging and trading baskets, among which balanced baskets have attracted significant attention in recent years. Unlike PCA-style methods, balanced baskets spread risk or exposure across its constituents without requiring a change of basis. Practitioners typically prefer balanced baskets because their output can be understood in intuitive terms.

In this paper we will differentiate between the goal of hedging and the goal of trading. In the first instance, the basket is formed to reduce the investor's risk or exposure to one of its legs, or a subset of them. In the second instance, the investor would like to acquire risk or exposure to each and every of its legs (or subsets of them) in a balanced way. Although hedging baskets may seem the opposite to trading baskets, both concepts are intimately related and both can be computed using similar procedures.

López de Prado and Leinweber (2012) reviewed the literature on hedging methods. Among the methods they studied are Equal-Risk Contribution (ERC), Maximum Diversification Ratio (MDR) and Mini-Max Subset Correlation (MMSC). The three are static methods that attempt to balance the risk or exposure of the basket among its constituents. MDR solves a "hedging" problem, while MMSC and ERC can be applied to solve both, a "hedging" as well as a "trading" problem.

Maillard, Roncalli and Teiletche (2009) and Demey, Maillard and Roncalli (2010) gave a formal definition of ERC. Previous descriptions can be found in Qian (2005, 2006) under the term *"risk parity"*, Booth and Fama (1992), and earlier authors. This procedure attempts to balance the contribution of risk for each of the basket's legs. Empirical studies of its performance against alternative weighting schemes can be found in Neukirch (2008), DeMiguel, Garlappi and Uppal (2009) and Hurst, Johnson and Ooi (2010). Most authors impose the constraints that all weights must be positive and add up to one, because they have in mind an asset allocation application. Our analysis is free of such constraints because we would like to discuss the problem of constructing a basket in general terms, rather than focusing on a particular use. This concept's popularity is demonstrated by the many institutional asset managers offering ERC-weighted funds: PanAgora Asset Management, Bridgewater Associates, AQR Capital, Aquila Capital, Invesco, First Quadrant, Putnam Investments, ATP, Barclays Global Investors, Mellon Capital Management, State Street Global Advisors… to cite only a few.

MDR was proposed by Choueifaty and Coignard (2008) and Choueifaty, Froidure and Reynier (2011). Their goal is to maximize a "diversification ratio" that effectively balances the basket's exposure to each leg, measured in terms of correlation. Like in the ERC case, these authors also incorporate constraints characteristic of an asset allocation framework, which we will obviate in this study for the sake of generality.

MMSC was introduced by López de Prado and Leinweber (2012). This procedure balances the exposure of the basket, not only to each leg (like MDR) but also to any subset of legs. The motivation is to reduce the basket's vulnerability to structural breaks, i.e. when a subset receives

a shock that does not impact all of the legs. In a basket of two instruments, MMSC coincides with MDR, since the only subsets are the legs themselves. Furthermore, we will see that when only two instruments are considered, ERC, MDR and MMSC give the same solution. However, the three procedures exhibit substantial differences whenever we are dealing with baskets of more than two instruments.

The three procedures are theoretically sound. The purpose of this study is not to invalidate or criticize any of them, but to evidence the differences and properties associated with each solution. A second goal of this paper is to provide efficient algorithms for the calculation of ERC, MDR and MMSC, which cannot be found in the referred literature. A third, ancillary goal, is to provide a procedure for reducing the dimension of a covariance matrix to a number that makes these procedures computationally feasible. We believe that our *Covariance Clustering* method has important applications for the management of risks in large portfolios of highly correlated instruments or funds.

The rest of the paper is organized as follows: Section 2 discusses the hedging problem in a two-dimensional framework. Section 3 evidences the qualitative difference between working in two dimensions and dealing with three or more. Section 4 extends our "hedging" analysis to the problem of computing "trading baskets." Section 5 summarizes our conclusions. Appendix 1 derives a numerical procedure for the calculation of ERC baskets. Appendix 2 presents a codification of that algorithm in Python. Appendices 3 and 4 do the same in the context of MMSC and MDR baskets. Appendix 5 describes the Covariance Clustering method.

2. THE TWO-DIMENSIONAL HEDGING PROBLEM

Suppose that a portfolio manager wishes to hedge her position of 1,000 S&P Midcap 400 E-mini futures contracts (Bloomberg code "FA1 Index") using S&P 500 E-mini futures (Bloomberg code "ES1 Index"). The relevant covariance and correlation matrices on daily market value dollar changes (ΔP_i) are shown in Figure 1. As expected, we can appreciate a high and positive codependence between these two products, with an estimated correlation coefficient approaching 0.95.

[FIGURE 1 HERE]

Given some weights ω and a covariance matrix V of market value dollar changes, we can compute the variance of market value changes of the resulting basket B of n constituents as

$$\sigma_{\Delta B}^2 = \omega' V \omega = \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_i \; \sigma_{i,j} \omega_j \tag{1}$$

where $\sigma_{i,j}$ is the covariance between ΔP_i and ΔP_j. One option would be to compute the vector ω_{MV} that minimizes the basket's variance,

$$\omega_{MV} = \arg \min_{\substack{\omega \\ \omega_1 = h_1}} \; \sigma_{\Delta B}^2 \tag{2}$$

for which a general solution can be found in López de Prado and Leinweber (2012). In our particular case, $n=2$ and our holdings of FA1 Index are set fix to be $h_1 = 1,000$, so the only free weight is ω_2. Deriving,

$$\frac{\partial \sigma_{\Delta B}^2}{\partial \omega_2} = 2h_1\sigma_{12} + 2\omega_2\sigma_2^2 \tag{3}$$

$$\frac{\partial^2 \sigma_{\Delta B}^2}{\partial \omega^2} = 2\sigma_2^2 > 0$$

Because $\frac{\partial^2 \sigma_{\Delta B}^2}{\partial \omega^2} > 0$, applying the first order condition ($\frac{\partial \sigma_{\Delta B}^2}{\partial \omega_2} = 0$) will lead to a minimum at the point $\omega_{MV} = h_1\left(1, -\frac{\sigma_{1,2}}{\sigma_2^2}\right) \approx (1000, -1467.85)$, with $\sigma_{\Delta B}^2 \approx 299,709.02$. The solution matches the coefficients of an Ordinary Least Squares (OLS) regression, which should have been expected, since the regression's objective function coincides with Eq. (2) for the case $n=2$.

Moulton and Seydoux (1998), Scherer (2010) and Clarke, de Silva and Thorley (2011) provide practical examples of OLS and MV solutions. Although ubiquitous, the solutions provided by these procedures exhibit a few undesirable traits. For example, note that if we had switched contracts, the alternative second weight would have been $\widetilde{\omega}_2 = -\frac{\widetilde{\sigma}_{1,2}}{\widetilde{\sigma}_2^2} = -\frac{\sigma_{1,2}}{\sigma_1^2}$, and $\widetilde{\omega}_2 \neq \frac{1}{\omega_2}$ unless $\sigma_1^2 = \sigma_2^2$. So the ordering of the instruments introduces some arbitrariness to this solution. Also, the risk contributed by each of the basket's constituents is not equal. In order to evaluate that, we need first to formalize the concept of Contribution to Risk.

2.1. CONTRIBUTION TO RISK
The marginal contribution to the basket's variance from a leg i is

$$\frac{\partial \sigma_{\Delta B}}{\partial \omega_i} = \frac{1}{2}(\sigma_{\Delta B}^2)^{-1/2}\frac{\partial \sigma_{\Delta B}^2}{\partial \omega_i} = \frac{\sigma_{\Delta B,\Delta P_i \omega_i}}{\omega_i \sigma_{\Delta B}} = \frac{\sigma_i \sum_{j=1}^n \omega_j \sigma_j \rho_{i,j}}{\sigma_{\Delta B}} \tag{4}$$

where $\sigma_{\Delta B,\Delta P_i \omega_i}$ is the covariance between changes in the basket's dollar value and changes in the i-th leg's dollar value. Because volatility is a homogeneous function of degree 1, Euler's theorem allows us to write

$$\sigma_{\Delta B} = \sum_{i=1}^n \omega_i \frac{\partial \sigma_{\Delta B}}{\partial \omega_i} = \frac{1}{\sigma_{\Delta B}}\sum_{i=1}^n [V\omega \circ \omega]_i \tag{5}$$

where $\circ$ represents a Hadamard product. We can define Contribution to Risk (CtR) as

$$CtR_i = \frac{\partial \sigma_{\Delta B}}{\partial \omega_i}\frac{\omega_i}{\sigma_{\Delta B}} = \frac{[V\omega \circ \omega]_i}{\omega'V\omega} = \frac{\omega_i \sigma_i \sum_{j=1}^n \omega_j \sigma_j \rho_{i,j}}{\sigma_{\Delta B}^2} \tag{6}$$

and obviously $\sum_{i=1}^{n} CtR_i = 1$. One hedging option would be to estimate the Equal-Risk Contribution vector (ERC), such that $CtR_i = CtR_j, \forall i,j$. In our *n=2* case, that occurs when

$$h_1^2 \sigma_1^2 + h_1 \sigma_1 \omega_2 \sigma_2 \rho_{1,2} = \omega_2^2 \sigma_2^2 + \omega_2 \sigma_2 h_1 \sigma_1 \rho_{1,2}$$

$$\omega_2 = -h_1 \frac{\sigma_1}{\sigma_2}$$

(7)

which leads to $\omega_{ERC} \approx (1000, -1552.48)$. Unlike in the OLS solution, the ERC solution is not affected by the order of the instruments. The risk of ERC's hedging basket is $\sigma_{\Delta B}^2 \approx 303,879.05$. That is very close to the minimum risk of the MV-OLS solution, with the advantage that with ERC we get $CtR_1 = CtR_2 = \frac{1}{2}$. That's not the case of $\omega_{MV} \approx (1000, -1467.85)$, for which we obtain $CtR_1 = 1$ and $CtR_2 = 0$. So although the MV-OLS solution has reduced risk from 920,304.74 to a minimum 299,709.02, the first leg is still responsible for the entirety of the risk. In order to understand why, we need to introduce the concept of Correlation to Basket (CtB).

2.2. CORRELATION TO BASKET
The correlation of each constituent to the overall basket can be computed as

$$\sigma_{\Delta P_i \omega_i, \Delta P_j \omega_j} = \omega_i \sigma_{i,j} \omega_j = \omega_i \sigma_i \omega_j \sigma_j \rho_{i,j}$$

(8)

$$\sigma_{\Delta B, \Delta P_i \omega_i} = \sum_{j=1}^{n} \sigma_{\Delta P_i \omega_i, \Delta P_j \omega_j} = \omega_i \sigma_i \sum_{j=1}^{n} \omega_j \sigma_j \rho_{i,j}$$

(9)

$$CtB_i = \rho_{\Delta B, \Delta P_i \omega_i} = \frac{\sigma_{\Delta B, \Delta P_i \omega_i}}{\sigma_{\Delta B} \sigma_{\Delta P_i \omega_i}} = Sgn(\omega_i) \frac{\sum_{j=1}^{n} \omega_j \sigma_j \rho_{i,j}}{\sigma_{\Delta B}} = Sgn(\omega_i) \rho_{\Delta B, \Delta P_i}$$

(10)

$Sgn(\omega_i)$ arises from simplifying $\frac{\omega_i}{\sqrt{\omega_i^2}}$. López de Prado and Leinweber (2012) show how to compute a general hedging basket such that $CtB_i = CtB_j, \forall i,j$, which the authors call MMSC (for Mini-Max Subset Correlations). In our particular *n=2* case, we expect the hedge to be $\omega_2 < 0$ since $\rho_{1,2} > 0$, so the equal exposure is achieved at

$$h_1 \sigma_1 + \omega_2 \sigma_2 \rho_{1,2} = -\omega_2 \sigma_2 - h_1 \sigma_1 \rho_{1,2}$$

$$\omega_2 = -h_1 \frac{\sigma_1}{\sigma_2}$$

(11)

which means that the MMSC and ERC solutions coincide when *n=2*. There is a connection between CtR and CtB, as evidenced by Eq. (6) and Eq. (10), specifically,

$$CtR_i = \frac{|\omega_i|\sigma_i}{\sigma_{\Delta B}} CtB_i \tag{12}$$

From this expression we deduce that an ERC basket will match the MMSC solution when $|\omega_i|\sigma_i = |\omega_j|\sigma_j$. This condition is clearly met when *n=2* (see Eq. 11), however that is not generally true when *n>2*.

For $\omega_{MMSC} \approx (1000, -1552.48)$, we obtain $CtB_1 = CtB_2 \approx 0.17$, while for $\omega_{MV} \approx (1000, -1467.85)$ we obtain $CtB_1 \approx 0.33$ and $CtB_2 = 0$. So what the MV-OLS solution has done is to add an "ES1 Index" leg that is orthogonal to the basket. The risk from ω_{MV} is slightly lower (in fact, the lowest), however it is still concentrated in the first leg ($CtR_1 = 1$). These are good reasons for favouring MMSC or ERC solutions over MV or OLS.

2.3. SPECTRAL DECOMPOSITION

A change of basis will allow us to understand geometrically the previous analysis. Because a covariance matrix *V* is squared and symmetric, its eigenvector decomposition delivers a set of real valued orthonormal vectors that we can use to plot the MV-OLS and the MMSC-ERC solutions. Applying the Spectral Theorem,

$$V = W\Lambda W' \tag{13}$$

where Λ is the eigenvalues matrix, *W* is the eigenvectors matrix, and W' denotes its transpose. Λ is a diagonal matrix and the columns of *W* are orthogonal to each other, i.e. $W' = W^{-1}$. For convenience, we reorder columns in *W* and Λ so that $\Lambda_{i,i} \geq \Lambda_{j,j}, \forall j > i$. The *i*-th *principal component* is defined as a portfolio with the holdings listed in the *i*-th column of *W*. Looking at the above equation, Λ can then be interpreted as the covariance matrix between the principal components characterized by the columns of *W*. The factor loadings vector $f_\omega = W'\omega$ gives us the projection of ω into this new orthogonal basis. This can be verified from $\sigma_{\Delta B}^2 = \omega'V\omega = \omega'W\Lambda W'\omega = f_\omega'\Lambda f_\omega$. The product $f_I = W'I$, where *I* represents the identity matrix, gives us the directions of the old axes in the new basis.

Going back to our original example,

$$W = \begin{bmatrix} 0.84646 & -0.53245 \\ 0.53245 & 0.84646 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} 1171422.026 & 0 \\ 0 & 26946.176 \end{bmatrix} \tag{14}$$

For the initial position $\omega_0 \approx \begin{bmatrix} 1000 \\ 0 \end{bmatrix}$, the factor loadings are $f_{\omega_0} \approx \begin{bmatrix} 846.46203 \\ -532.44908 \end{bmatrix}$. It becomes evident that ω_0 is not well diversified, because the f_{ω_0} vector is not pointing in the direction of the second orthogonal component, which is the one with least variance. If we adopted the MV-OLS solution, the resulting basket would take a direction $f_{\omega_{MV}} \approx \begin{bmatrix} 64.90733 \\ -1774.9273 \end{bmatrix}$ in the orthonormal axes. That is a noticeable improvement over ω_0, as ω_{MV} is shown to be much closer to the second one orthogonal component, thus it remains little diversifiable risk in terms of these

two instruments ("FA1 Index" and "ES1 Index"). And yet the MMSC-ERC solution is better diversified, with $f_{\omega_{MMSC}} \approx \begin{bmatrix} 19.84512 \\ -1846.56505 \end{bmatrix}$.

The first component is typically associated with market risk, of which ω_{MMSC} exhibits the least. The MMSC basket is almost completely associated with spread risk, which is best characterized by the second component. López de Prado and Leinweber (2012) show how to compute in general a hedging portfolio with zero exposure to the first components. For the $n=2$ case, it means that

$$
f_\omega = W' \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \dfrac{\sigma_{\Delta B}}{\sqrt{\Lambda_{2,2}}} \end{bmatrix}
$$

$$
\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = W \begin{bmatrix} 0 \\ \dfrac{\sigma_{\Delta B}}{\sqrt{\Lambda_{2,2}}} \end{bmatrix} = \begin{bmatrix} W_{1,2} \\ W_{2,2} \end{bmatrix} \dfrac{\sigma_{\Delta B}}{\sqrt{\Lambda_{2,2}}}
$$

(15)

and re-scaling to meet the condition that $\omega_1 = h_1$, we obtain that $\omega_{PCA} = h_1 \left(1, \dfrac{W_{2,2}}{W_{1,2}} \right)$. In our example, this means that $\omega_{PCA} \approx (1000, -1589.75)$. Then, $f_{\omega_{PCA}} \approx \begin{bmatrix} 0 \\ -1878.11386 \end{bmatrix}$, which indicates that we are perfectly hedged against the most volatile of the orthogonal component (typically known as the "market component"), as expected. Figure 2 graphs these vectors.

[FIGURE 2 HERE]

The columns of *W* (also called Principal Components) represent linear combinations of the securities that happen to be orthogonal to each other, sorted descending in terms of the variance associated with that direction, with a unit length. In this PCA approach, we are hedged against all risk coming from the first *n-1* components (with the largest variance), and fully concentrated on the *n*-th component (with the smallest variance). One caveat of the PCA approach is that the interpretation of the *n*-th component is not necessarily intuitive in terms of the basket constituents. For example, in the ERC approach we knew that each leg contributed equal risk, and in the MMSC approach that the basket had equal exposure to each leg. That direct connection between the solution and the basket constituents is missing in the PCA approach, because of the change of basis. This is generally perceived as a drawback, particularly in highly-dimensional problems. MMSC is an appealing alternative to PCA because MMSC searches for a basket as orthogonal as possible to the legs, without requiring a basis change (like PCA). So although MMSC's solution is close to PCA's, it can still be linked intuitively to the basket's constituents. Understanding how this is done beyond the two-dimensional framework requires us to introduce the concept of subset correlation.

3. HEDGING BEYOND TWO DIMENSIONS

In the previous section we argued in favor of MMSC and ERC over alternative hedging approaches such as MV or PCA. The problem with MV's solution was that risk contribution (measured in terms of CtR) and exposure (measured in terms of CtB) of the basket's constituents is typically unbalanced. The problem with PCA was that the solution cannot be immediately (or for that matter, intuitively) understood in terms of the basket's constituents. In this section we will examine the distinct features of MMSC and ERC beyond the two-dimensional case.

The previous example was simplified by the fact that we were only considering two dimensions. As we will see next, there is a qualitative difference between working in two dimensions and working with three or more. Suppose that a portfolio manager wishes to hedge her position of 1,000 S&P Midcap 400 E-mini futures contracts ("FA1 Index") using two other instruments: S&P 500 E-mini futures ("ES1 Index") and DJAI E-Mini futures ("DM1 Index"). The relevant covariance and correlation matrices of daily market value (dollar) changes are shown in Figure 3.

[FIGURE 3 HERE]

As expected, we can appreciate a high and positive codependence between these three products. It is of course desirable to hedge our position using instruments highly correlated to it (about 0.95 correlation against "ES1 Index", and 0.91 correlation against "DM1 Index"). Unfortunately, that comes at the cost of having to deal with a similarly high correlation between the hedging instruments (about 0.98 correlation between "ES1 Index" and "DM1 Index"). This poses a problem because there may be an overlap between the hedges, which was not present in the two-dimensional case.

3.1. THE EQUAL-RISK CONTRIBUTION (ALIAS, RISK PARITY) SOLUTION

We will illustrate this point by computing the ERC basket, which solves the problem

$$\omega_{ERC} = \left\{ \omega_i \left| \frac{\omega_i \sigma_i \rho_{\Delta B, \Delta P_i}}{\sum_{j=1}^{n} \omega_j \sigma_j \rho_{\Delta B, \Delta P_j}} = \frac{1}{n}, \forall i \right. \right\} \tag{16}$$

Appendix 1 provides the details of this calculation, for any n dimensions, and Appendix 2 offers an algorithm coded in Python which computes the ERC basket.

[FIGURE 4 HERE]

Figure 4 reports the results of applying this algorithm to the input variables in Figure 3. A first problem with this result is the uneven correlations to the basket (CtB). The "ES1 Index-DM1 Index" subset will dominate the performance of the hedge, with its 0.26 correlation to the basket. This could have potentially serious consequences should there exist a correlation break between "ES1   Index"   and   "DM1   Index"   on   one   hand   and FA1 Index" on the other. A second problem is that the solution itself is not unique. Figure 5 presents an alternative solution for which also $CtR_i \approx \frac{1}{3}, \forall i$, with unacceptably high values of $CtB_2 > 0.99$. We would of course reject this alternative solution out of common sense, however

it would be better to rely on a procedure that searches for reasonable hedges, if possible with unique solutions.

[FIGURE 5 HERE]

In conclusion, ERC does not necessarily deliver a unique and balanced (exposure-wise) solution when *n>2*.

3.2. DIVERSIFIED RISK PARITY

Building on an idea of Meucci (2009), Lohre, Neugebauer and Zimmer (2012) and Lohre, Opfer, and Ország (2012), proposed a very interesting variation of ERC that they called "Diversified Risk Parity" (DRP). It computes the allocations such that the contribution to risk from every principal component is equal. In a nutshell, DRP is like ERC, but computed on the principal components instead of the actual instruments. Like PCA, DRP also requires a spectral decomposition, and thus a change of basis. We cannot strictly classify it as a balanced basket, however we will derive its calculation for illustration purposes. Our DRP solution differs from Lohre, Neugebauer and Zimmer (2012) in two aspects: First, we are not imposing the asset allocation constraints (non-negativity, add to one), because they are not relevant in a hedging framework. Second (and as a consequence of the first), our solution is analytical, while theirs is numerical.

Recall from Section 2.3 that the eigenvectors of a symmetric matrix are orthonormal, thus $W'W = I$ and $\Lambda_{i,j} = 0, \forall i \neq j$. The correlation between any two principal components is $\frac{\Lambda_{i,j}}{\sqrt{\Lambda_{i,i}\Lambda_{j,j}}} = 0$. This implies that, given some factor loadings $f_\omega = W'\omega$, the basket's variance can be simply decomposed as $\sigma_{\Delta B}^2 = \sum_{i=1}^n [f_\omega]_i^2 \Lambda_{i,i}$, where $[f_\omega]_i$ is the i-th element of the factor loading's vector. We conclude that the contribution to risk of the i-th principal component is

$$\widetilde{CtR}_i = \frac{[f_\omega]_i^2 \Lambda_{i,i}}{\sum_{j=1}^n [f_\omega]_j^2 \Lambda_{j,j}} \tag{17}$$

Suppose that $[f_\omega]_i = \sqrt{\frac{\Lambda_{1,1}}{\Lambda_{i,i}}}, \forall i$. Then, $\widetilde{CtR}_i = \frac{\frac{\Lambda_{1,1}}{\Lambda_{i,i}}\Lambda_{i,i}}{\Lambda_{1,1}\sum_{j=1}^n \frac{1}{\Lambda_{j,j}}\Lambda_{j,j}} = \frac{1}{n}$, which is the ERC solution on factor loadings f_ω. But f_ω are loadings in the new basis (of principal components), and we still need to derive the weightings in the old basis (of actual instruments). Since $f_\omega = W'\omega$ and $W' = W^{-1}$, these can be computed as

$$\omega_{DRP} = W'^{-1}f_\omega = Wf_\omega \tag{18}$$

In our particular example, that would lead to the solution shown in Figure 6, and again we obtain a result that, although mathematically correct, it defies common sense. $\widetilde{CtR}_i = \frac{1}{3}, \forall i$, and yet the CtR and CtB (in terms of the instruments) exhibit a large concentration on "DM1 Index". Another inconvenience of DRP, which in fact is shared by all PCA-style approaches, is their lack of robustness: A change in just one observation may produce a last eigenvector that spans in a

completely different direction, and because DRP equally distributes risk among all principal components, the impact on ω_{DRP} will be dramatic. This makes PCA-like baskets in general, and DRP in particular, vulnerable to structural breaks and outliers.

[FIGURE 6 HERE]

3.3. THE MAXIMUM-DIVERSIFICATION SOLUTION

Choueifaty and Coignard (2008) compute the vector of holdings that Maximize the *Diversification Ratio* (MDR), as defined by

$$\omega_{MDR} = \arg\max_{\omega} \quad \frac{\sum_{i=1}^{n} \omega_i \sigma_i}{\sum_{i=1}^{n} \omega_i \sigma_i \rho_{\Delta B, \Delta P_i}} \tag{19}$$

This diversification ratio is the ratio of weighted volatilities divided by the basket's volatility. MDR is an intuitive method that penalizes the risk associated with cross-correlations, as they are accounted by the denominator but absent in the numerator of the maximized ratio. Choueifaty, Froidure and Reynier (2011) show that the correlations of each leg to the MDR hedging basket are minimized and made equal.[1] Figure 7 presents the MDR solution to our example. $CtB_i \approx 0.08, \forall i$, and there is no way we can make CtB smaller for all legs. Now CtR values make more sense, because the leg with a negative weight is responsible for almost ½ of the total risk, with the other half going to the legs with a positive weight. Among the "long" legs the risk is not equally spread, because the correlation between "DM1 Index" and "ES1 Index" (about 0.98) is greater than the correlation between "FA1 Index" and "ES1 Index" (about 0.95). So the MDR result seems intuitive and preferable to the ERC result.

[FIGURE 7 HERE]

And yet, if we dig deeper into this MDR result we will find something not entirely satisfactory. The problem is, there are subsets of instruments that, combined, exhibit greater correlation to the overall basket. Figure 7 shows that, even though the exposure is perfectly balanced at the leg level, this hedging basket's performance may still be dominated by some groups of legs. In particular, there is an approx. 0.36 correlation between the subset made of ("ES1 Index", "DM1 Index") and the overall basket. Like in the ERC case, that could be a source of losses should there be a correlation break between large-cap and mid-cap stocks, as witnessed most recently during the 2008 financial crisis. Furthermore, Choueifaty and Coignard (2008) acknowledge that the solution may not be unique or robust, particularly with ill-conditioned covariance matrices. Adding some structure to the optimization program would alleviate these problems.

Succinctly, although MDR is to some extent preferable to ERC, it does not address the problems of uniqueness of solution and balanced exposure of subsets of legs to the overall basket.

---

[1] As we will show in the next Section, this is equivalent to a MMSC where only subsets of one instrument are taken into consideration.

3.4. THE MINI-MAX SUBSET CORRELATION SOLUTION

We denote *subset correlation* as the correlation of a subset of instruments to the overall basket. MMSC's goal is to prevent that any leg or subset of legs dominates the basket's performance, as measured by its subset correlations. This additional structure adds the robustness and uniqueness of solution that were missing in ERC and MDR. MMSC baskets are also more resilient to *structural breaks*, because even if a subset of legs breaks it correlation to the rest of the basket, MMSC is the solution that has minimized its dependency to any particular subset of legs. When we were dealing with 2 instruments, the only subsets were the instruments themselves, so the only subset correlations were the CtBs. The MMSC solution coincided with the MDR solution. But now that *n=3*, we can compute correlations to 6 subsets (the 3 single legs plus the 3 possible pairs of legs), and we need to distinguish between both procedures. The solution can be characterized as

$$\omega_{MMSC} = \arg \quad \min_{\omega} \left\{ \max_{i} |\rho_{\Delta B, \Delta P_i \omega_i}| \right\} \tag{20}$$

where $i=1,...,N$ subsets and $N = \sum_{i=1}^{n-1} \binom{n}{i} = 2(2^{n-1} - 1)$ is the number of subsets (excluding the empty set and the full set). $N>n$ when $n>2$. As n grows, we will have many more subsets (N) than instruments (n). Ideally we would like to minimize all subset correlations and bring them as close to each other as possible, thus the name MMSC (Mini-Max Subset Correlations). Appendix 3 presents an algorithm for computing the MMSC solution for any dimension, and Appendix 4 provides the code in Python. Figure 8 shows the results for our example. The greatest correlation of any subset to the overall basket is about 0.185, significantly lower than in the ERC and MDR cases.

[FIGURE 8 HERE]

$CtR_1 \approx \frac{1}{2}$ seems to point at a concentration of risk in the "FA1 Index" leg. This is an artifact of computing CtR when one of the legs has low correlation to the basket. In this instance, CtR will not be able to accurately split risk among the instruments. From Appendix 1 we know that

$$\frac{\partial CtR_i}{\partial \omega_i} = \frac{\omega_i \sigma_i^2}{\sigma_{\Delta B}^2} \left(1 - 2\rho_{\Delta B, \Delta P_i \omega_i}^2 \right) + \frac{\sigma_i \rho_{\Delta B, \Delta P_i \omega_i}}{\sigma_{\Delta B}} \tag{21}$$

$\frac{\partial CtR_2}{\partial \omega_2} \approx \frac{\omega_2 \sigma_2^2}{\sigma_{\Delta B}^2}$ because $\rho_{\Delta B, \Delta P_2 \omega_2}^2 \approx 0$. $\frac{\sigma_i}{\sigma_{\Delta B}} > 1$, because this is a hedging basket. Thus, a very small change in the weights could transfer a substantial amount from CtR_1 to CtR_2, which is counter-intuitive. We can also see that, despite the high CtR_1 value, CtB_1 is virtually the same as the correlation of the subset formed by ("FA1 Index", "ES1 Index") to the basket, or the correlation of the subset ("ES1 Index", "DM1 Index") to the basket. In contrast, CtB is more stable, with $\frac{\partial \rho_{\Delta B, \Delta P_2 \omega_2}}{\partial \omega_2} \approx \frac{\sigma_2}{\sigma_{\Delta B}}$. The correlation to the basket does not have a factor ω_i or a second power on $\frac{\sigma_i}{\sigma_{\Delta B}} > 1$, thus it is more resilient to small changes in ω_2 (another reason why MMSC will tend to be more robust than ERC):

$$\frac{\partial \rho_{\Delta B, \Delta P_i \omega_i}}{\partial \omega_i} = \frac{\sigma_i}{\sigma_{\Delta B}} \left(1 - \rho_{\Delta B, \Delta P_i \omega_i}^2\right) \qquad (22)$$

Summarily, MMSC provides the basket with the most balanced exposure to any of its legs or subsets of legs. The solution is also unique and more robust, virtue of the 6 conditions that have been met by only 3 weights. The fact that $N \gg n$ as n grows provides a competitive advantage for users with access to high performance computing (HPC) facilities, who can deploy a solution unavailable to investors with limited computational power. When $n>30$, the computations involved would present a challenge for today's supercomputers. However, that obstacle could be surmounted by clustering the legs into highly correlated blocks, as proposed in Appendix 5. A second alternative is to limit the maximum size of the subsets evaluated, which can be done through the parameter *maxSubsetSize* in the Python code provided in Appendix 4.

3.5. GEOMETRIC INTERPRETATION

As we did in the two-dimensional case, we can understand the previous three-dimensional results in geometric terms. For the initial position $\omega_0 = \begin{bmatrix} 1000 \\ 0 \\ 0 \end{bmatrix}$, the factor loadings are $f_{\omega_0} \approx$ $\begin{bmatrix} 774.88384 \\ -620.81023 \\ 118.95250 \end{bmatrix}$. As expected, ω_0 is not well diversified, because the f_{ω_0} vector is nowhere close to the orthogonal directions with least variance. ERC is better diversified, with $f_{\omega_{ERC}} \approx$ $\begin{bmatrix} 26.23808 \\ -558.67818 \\ 5320.06831 \end{bmatrix}$. For MDR, we get $f_{\omega_{MDR}} \approx \begin{bmatrix} 14.14535 \\ -388.45011 \\ 6287.25928 \end{bmatrix}$, and for MMSC we get $f_{\omega_{MMSC}} \approx$ $\begin{bmatrix} 33.59253 \\ -691.79627 \\ 4577.41954 \end{bmatrix}$. So it would appear as if, in this particular example, MMSC did a worse job than MDR and ERC. Figure 9 tells us that this is not the case.

[FIGURE 9 HERE]

The first table reports the factor loadings of the three balanced hedging baskets to the first principal component. Although $f_{\omega_{MMSC}}(1) \approx 33.59253$ is slightly greater than for the other baskets, when it comes to the subsets we find that $f_{\omega_{MMSC}}$ is generally the least exposed. This is an important feature because, in the case of a *structural break*[2], MMSC's hedging basket will tend to be the least impacted. For instance, suppose that instruments 1 and 2 suffer a shock in the direction of the first principal (or market) component, without affecting instrument 3 (hence the structural break). In that case, the subset formed by instruments 1 and 2 will drive the full performance of the basket. As we can see, in that scenario MMSC would be the basket least exposed to the shock (-1023.5954), followed by ERC (-1260.31618) and then MDR (-1569.97131). The second and third tables in Figure 9 tell us that a similar argument can be made for most shocks in the direction of the second and third principal components. This feature is the result of minimizing the correlation of any subset to the overall basket.

---

[2] In this context, we speak of a "structural break" with the meaning of a change in the direction of a principal component which does not affect all instruments in the basket.

4. TRADING BASKETS

As we have seen earlier, a hedging basket attempts to minimize the exposure to any of its constituents. In contrast, a trading basket tries to determine the weightings such that the exposure is maximized. With that difference in mind, the problem is again how to determine a basket with balanced exposures, i.e. that no particular leg or subset of legs is responsible for the overall basket's performance.

As discussed earlier, MDR's goal is to maximize diversification by minimizing all CtB. This is consistent with a hedging problem. As originally formulated, it cannot be used to compute a trading basket.

The ERC procedure can sometimes deliver a trading basket by chance. The result reported in Figure 5 happened to be a trading basket, even though we were searching for a hedging basket. This is because ERC does not have a way to control for those two different objectives. ERC may converge to one or the other depending on the initial seed (e.g., a vector of ones in the case of Figure 5). This problem could be circumvented by trying different seeds. As expected, the resulting ERC trading basket for our example is balanced in terms of contribution to risk (CtR).

When used to compute a trading basket, MMSC will maximize the minimum subset correlation (as opposed to minimizing the maximum subset correlation). In every iteration, it will push the lowest subset correlation to a higher value, thus raising the average correlation, until there is no way to increase any correlation without reducing another.

$$\omega = \arg \quad \max_{\omega} \left\{ \min_i \left| \rho_{\Delta B, \Delta P_i \omega_i} \right| \right\} \tag{23}$$

where $i=1,...,N$, and $N = \sum_{i=1}^{n-1} \binom{n}{i} = 2(2^{n-1} - 1)$ is the number of subsets. Figure 10 shows the trading basket computed by MMSC on the same three instruments used in the previous section. MMSC recognizes that a linear combination of "FA1 Index" and "DM1 Index" makes "ES1 Index" redundant, thus giving $\omega_2 \approx 0$. As a result, MMSC spreads risk equally between "FA1 Index" and "DM1 Index", rather than between three instruments (including a redundant position on "ES1 Index"), like the ERC approach did. Perhaps more important is that MMSC actively searches for this trading basket, rather than arriving at it thanks to a fortunate seed. Appendix 4 provides the Python code that computes MMSC hedging and trading baskets, depending on the user's preference.

[FIGURE 10 HERE]

5. CONCLUSIONS

PCA is a good theoretical option for computing hedging baskets. One drawback of the PCA approach is that the interpretation of the n-th component is not necessarily intuitive in terms of the basket constituents. This is due to the change of basis involved in PCA's eigen-decomposition of variance, and for $n>3$ it is difficult to visualize the intuition behind the

resulting eigenvectors. A case in point was the solution delivered by the DRP basket which, although mathematically correct, it defied common sense.

For this reason, practitioners typically rely on approaches that allow them to relate the basket's weightings to the statistical properties of the basket's constituents. Such is the purpose of balanced hedging baskets, which are characterized by spreading risk or exposure across the baskets' constituents, so that the combined risk is not only minimal but also well diversified. Although the solution is not a perfect hedge (i.e., a minimum variance portfolio orthogonal to the main principal components), there is no change of basis involved and therefore the basket can be understood in terms of its constituents.

How well risk is spread is measured by the Contribution to Risk (CtR) from each leg. Similarly, how well exposure is spread is measured by each leg's Correlation to the Basket (CtB). Three methods have been proposed to compute balanced hedging baskets: Equal-Risk Contribution (ERC), Maximum Diversification (MDR) and Mini-Max Subset Correlation (MMSC). All three are theoretically sound and useful procedures, and the purpose of this paper is not to disqualify any of them but to examine the properties of their respective solutions.

ERC computes the vector of weightings such that the CtR of each leg is equal. Absent of constraints, the solution is not unique and it could lead to hedging baskets riskier than any of the individual holdings. This problem can be circumvented by trying alternative seeds, however the procedure *per se* does not control the outcome. Also, although the solution always delivers equal CtR per leg, the CtB could be very high for some legs, evidencing concentration of exposure.

MDR is equivalent to ERC, with the difference that instead of equalizing CtR, it attempts to achieve equal CtB per leg. Solutions tend to be more intuitive, however they are not unique. Another caveat is that, although each leg's CtB may be equal, there may be subsets of legs that are highly exposed, i.e. they will dominate the overall basket's performance.

MMSC computes a hedging basket for which not only CtB are as low as possible, but also the correlations of subsets of legs to the overall basket are minimized. The outcome is a basket which performance is not dominated by any of its constituents, individually or in subsets. This feature is important, because it makes the basket more resilient to structural breaks. Since the number of subsets of legs is necessarily greater than the number of legs, the system is over-determined: There may be no solution that equalizes all subset correlations, in which case MMSC computes the Mini-Max approximation. Another advantage of MMSC is that it can also be used to compute trading baskets. These are characterized by a vector of weightings such that the correlation of each leg or subset of legs to the overall basket is maximized (rather than minimized, like in the hedging case).

One caveat of MMSC is that, for very large baskets, the number of subsets can be enormous, and the calculation of its solution may require access to high performance computing (HPC) facilities, such as the NERSC facility at Lawrence Berkeley National Laboratory.[3] We hope that readers will find helpful the numerically efficient algorithms provided in the Appendices.

---

[3] Additional details are available at: http://www.nersc.gov/users/computational-systems/

A.1. COMPUTING THE ERC BASKET FOR *n*-INSTRUMENTS
A.1.1. TAYLOR'S EXPANSION

A second degree Taylor expansion of the CtR function takes the expression:

$$
\begin{aligned}
CtR_i(\omega_i + \Delta\omega_i) \\
= CtR_i(\omega_i) + \frac{\partial CtR_i(\omega_i)}{\partial \omega_i}\Delta\omega_i + \frac{1}{2}\frac{\partial^2 CtR_i(\omega_i)}{\partial \omega_i{}^2}(\Delta\omega_i)^2 \\
+ \sum_{k=3}^{\infty}\frac{1}{k!}\frac{\partial^k CtR_i(\omega_i)}{\partial \omega_i{}^k}(\Delta\omega_i)^k
\end{aligned}
\tag{24}
$$

So we need to compute an analytical expression for the first and second partial derivatives.

A.1.2. FIRST DERIVATIVE

From Eq. (6) we know that $CtR_i = \frac{\partial \sigma_{\Delta B}}{\partial \omega_i}\frac{\omega_i}{\sigma_{\Delta B}}$, thus

$$
\frac{\partial CtR_i}{\partial \omega_i} = \frac{\partial \left[\frac{\partial \sigma_{\Delta B}}{\partial \omega_i}\frac{\omega_i}{\sigma_{\Delta B}}\right]}{\partial \omega_i} = \frac{\partial^2 \sigma_{\Delta B}}{\partial \omega_i{}^2}\frac{\omega_i}{\sigma_{\Delta B}} + \frac{\partial \left[\frac{\omega_i}{\sigma_{\Delta B}}\right]}{\partial \omega_i}\frac{\partial \sigma_{\Delta B}}{\partial \omega_i}
\tag{25}
$$

We already know that $\frac{\partial \sigma_{\Delta B}}{\partial \omega_i} = \frac{\sigma_{\Delta B,\Delta P_i \omega_i}}{\omega_i \sigma_{\Delta B}}$ from Eq. (4). We need to compute $\frac{\partial^2 \sigma_{\Delta B}}{\partial \omega_i{}^2}$ and $\frac{\partial \left[\frac{\omega_i}{\sigma_{\Delta B}}\right]}{\partial \omega_i}$. From López de Prado and Leinweber (2012) we also know that $\frac{\partial \sigma_{\Delta B}}{\partial \omega_i} = \sigma_i \rho_{\Delta B,\Delta P_i \omega_i}$ and $\frac{\partial \rho_{\Delta B,\Delta P_i \omega_i}}{\partial \omega_i} = \frac{\sigma_i}{\sigma_{\Delta B}}\left(1 - \rho^2_{\Delta B,\Delta P_i \omega_i}\right)$, thus

$$
\frac{\partial^2 \sigma_{\Delta B}}{\partial \omega_i{}^2} = \frac{\partial \rho_{\Delta B,\Delta P_i \omega_i}}{\partial \omega_i}\sigma_i = \frac{\sigma_i^2}{\sigma_{\Delta B}}\left(1 - \rho^2_{\Delta B,\Delta P_i \omega_i}\right)
\tag{26}
$$

$$
\frac{\partial \left[\frac{\omega_i}{\sigma_{\Delta B}}\right]}{\partial \omega_i} = \frac{\sigma_{\Delta B} - \frac{\partial \sigma_{\Delta B}}{\partial \omega_i}\omega_i}{\sigma^2_{\Delta B}} = \frac{1}{\sigma_{\Delta B}} - \frac{\omega_i \sigma_i \rho_{\Delta B,\Delta P_i \omega_i}}{\sigma^2_{\Delta B}}
\tag{27}
$$

We therefore conclude that

$$
\begin{aligned}
\frac{\partial CtR_i}{\partial \omega_i} &= \frac{\omega_i \sigma_i^2}{\sigma^2_{\Delta B}}\left(1 - \rho^2_{\Delta B,\Delta P_i \omega_i}\right) + \left(\frac{1}{\sigma_{\Delta B}} - \frac{\omega_i \sigma_i \rho_{\Delta B,\Delta P_i \omega_i}}{\sigma^2_{\Delta B}}\right)\sigma_i \rho_{\Delta B,\Delta P_i \omega_i} \\
&= \frac{\omega_i \sigma_i^2}{\sigma^2_{\Delta B}}\left(1 - 2\rho^2_{\Delta B,\Delta P_i \omega_i}\right) + \frac{\sigma_i \rho_{\Delta B,\Delta P_i \omega_i}}{\sigma_{\Delta B}}
\end{aligned}
\tag{28}
$$

A.1.3. SECOND DERIVATIVE
Next, we will obtain the expression for

$$\frac{\partial^2 CtR_i}{\partial \omega_i{}^2} = \frac{\partial \left[\frac{\omega_i}{\sigma_{\Delta B}^2}\right]}{\partial \omega_i} \sigma_i^2 \left(1 - 2\rho_{\Delta B,\Delta P_i \omega_i}^2\right) - 4\rho_{\Delta B,\Delta P_i \omega_i} \frac{\partial \rho_{\Delta B,\Delta P_i \omega_i}}{\partial \omega_i} \frac{\omega_i \sigma_i^2}{\sigma_{\Delta B}^2} \qquad (29)$$
$$+ \frac{\partial \left[\frac{\rho_{\Delta B,\Delta P_i \omega_i}}{\sigma_{\Delta B}}\right]}{\partial \omega_i} \sigma_i$$

We need to determine the analytical expressions for $\dfrac{\partial\left[\frac{\omega_i}{\sigma_{\Delta B}^2}\right]}{\partial \omega_i}$ and $\dfrac{\partial\left[\frac{\rho_{\Delta B,\Delta P_i \omega_i}}{\sigma_{\Delta B}}\right]}{\partial \omega_i}$:

$$\frac{\partial \left[\frac{\omega_i}{\sigma_{\Delta B}^2}\right]}{\partial \omega_i} = \frac{1}{\sigma_{\Delta B}^2} - 2\frac{\omega_i \sigma_i \rho_{\Delta B,\Delta P_i \omega_i}}{\sigma_{\Delta B}^3} \qquad (30)$$

$$\frac{\partial \left[\frac{\rho_{\Delta B,\Delta P_i \omega_i}}{\sigma_{\Delta B}}\right]}{\partial \omega_i} = \frac{\frac{\partial \rho_{\Delta B,\Delta P_i \omega_i}}{\partial \omega_i} \sigma_{\Delta B} - \frac{\partial \sigma_{\Delta B}}{\partial \omega_i} \rho_{\Delta B,\Delta P_i \omega_i}}{\sigma_{\Delta B}^2} = \frac{\sigma_i}{\sigma_{\Delta B}^2}\left(1 - 2\rho_{\Delta B,\Delta P_i \omega_i}^2\right) \qquad (31)$$

with the conclusion that

$$\frac{\partial^2 CtR_i}{\partial \omega_i{}^2} = \left(\frac{1}{\sigma_{\Delta B}^2} - 2\frac{\omega_i \sigma_i \rho_{\Delta B,\Delta P_i \omega_i}}{\sigma_{\Delta B}^3}\right)\sigma_i^2\left(1 - 2\rho_{\Delta B,\Delta P_i \omega_i}^2\right)$$
$$- 4\rho_{\Delta B,\Delta P_i \omega_i}\frac{\sigma_i}{\sigma_{\Delta B}}\left(1 - \rho_{\Delta B,\Delta P_i \omega_i}^2\right)\frac{\omega_i \sigma_i^2}{\sigma_{\Delta B}^2} + \frac{\sigma_i}{\sigma_{\Delta B}^2}\left(1 - 2\rho_{\Delta B,\Delta P_i \omega_i}^2\right)\sigma_i$$
$$= \frac{\sigma_i^2}{\sigma_{\Delta B}^2} - \frac{\sigma_i^2}{\sigma_{\Delta B}^2}2\rho_{\Delta B,\Delta P_i \omega_i}^2 - 2\frac{\omega_i \sigma_i^3 \rho_{\Delta B,\Delta P_i \omega_i}}{\sigma_{\Delta B}^3} + 4\frac{\omega_i \sigma_i^3 \rho_{\Delta B,\Delta P_i \omega_i}^3}{\sigma_{\Delta B}^3} \qquad (32)$$
$$- 4\rho_{\Delta B,\Delta P_i \omega_i}\frac{\omega_i \sigma_i^3}{\sigma_{\Delta B}^3} + 4\rho_{\Delta B,\Delta P_i \omega_i}^3\frac{\omega_i \sigma_i^3}{\sigma_{\Delta B}^3} + \frac{\sigma_i^2}{\sigma_{\Delta B}^2} - \frac{\sigma_i^2}{\sigma_{\Delta B}^2}2\rho_{\Delta B,\Delta P_i \omega_i}^2$$
$$= 2\frac{\sigma_i^2}{\sigma_{\Delta B}^2}\left(1 - 2\rho_{\Delta B,\Delta P_i \omega_i}^2\right) - 2\rho_{\Delta B,\Delta P_i \omega_i}\frac{\omega_i \sigma_i^3}{\sigma_{\Delta B}^3}\left(2 - 3\rho_{\Delta B,\Delta P_i \omega_i}^2\right)$$

A.1.4. STEP SIZE
Finally, assuming $\sum_{k=3}^{\infty}\frac{1}{k!}\frac{\partial^k CtR_i(\omega_i)}{\partial \omega_i{}^k}(\Delta\omega_i)^k \approx 0$, we can replace these derivatives into Taylor's expansion:

$$\Delta CtR_i = CtR_i(\omega_i + \Delta\omega_i) - CtR_i(\omega_i)$$
$$= \frac{\partial CtR_i(\omega_i)}{\partial \omega_i}\Delta\omega_i + \frac{1}{2}\frac{\partial^2 CtR_i(\omega_i)}{\partial \omega_i{}^2}(\Delta\omega_i)^2 \qquad (33)$$

giving us the expression

$$\Delta CtR_i = \left[\frac{\omega_i \sigma_i^2}{\sigma_{\Delta B}^2}\left(1 - 2\rho_{\Delta B,\Delta P_i\omega_i}^2\right) + \frac{\sigma_i \rho_{\Delta B,\Delta P_i\omega_i}}{\sigma_{\Delta B}}\right]\Delta\omega_i$$
$$+ \left[\frac{\sigma_i^2}{\sigma_{\Delta B}^2}\left(1 - 2\rho_{\Delta B,\Delta P_i\omega_i}^2\right)\right. \tag{34}$$
$$\left. - \rho_{\Delta B,\Delta P_i\omega_i}\frac{\omega_i \sigma_i^3}{\sigma_{\Delta B}^3}\left(2 - 3\rho_{\Delta B,\Delta P_i\omega_i}^2\right)\right](\Delta\omega_i)^2$$

Let's define

$$a = \frac{\sigma_i^2}{\sigma_{\Delta B}^2}\left(1 - 2\rho_{\Delta B,\Delta P_i\omega_i}^2\right) - \rho_{\Delta B,\Delta P_i\omega_i}\frac{\omega_i \sigma_i^3}{\sigma_{\Delta B}^3}\left(2 - 3\rho_{\Delta B,\Delta P_i\omega_i}^2\right)$$
$$b = \frac{\omega_i \sigma_i^2}{\sigma_{\Delta B}^2}\left(1 - 2\rho_{\Delta B,\Delta P_i\omega_i}^2\right) + \frac{\sigma_i \rho_{\Delta B,\Delta P_i\omega_i}}{\sigma_{\Delta B}} \tag{35}$$
$$c = -\Delta CtR_i$$

Then, for $a \neq 0$ we will choose the smallest step size (to reduce the error due to Taylor's approximation, which grows with $|\Delta\omega_i|$):

$$\Delta\omega_i = \begin{cases} \dfrac{-b + \sqrt{b^2 - 4ac}}{2a} & if\ b \geq 0 \\ \dfrac{-b - \sqrt{b^2 - 4ac}}{2a} & if\ b < 0 \end{cases} \tag{36}$$

For $a = 0$, the solution coincides with a first degree Taylor approximation:

$$\Delta\omega_i = -\frac{c}{b} = \Delta CtR_i \left[\frac{\omega_i \sigma_i^2}{\sigma_{\Delta B}^2}\left(1 - 2\rho_{\Delta B,\Delta P_i\omega_i}^2\right) + \frac{\sigma_i \rho_{\Delta B,\Delta P_i\omega_i}}{\sigma_{\Delta B}}\right]^{-1} \tag{37}$$

A.2. THE ERC ALGORITHM

We know that the ω_{ERC} basket must verify that all Contributions to Risk are equal: $CtR_i = \frac{1}{n}, \forall i$. Thus, for any i we can compute the step size $\Delta\omega_i$ so that any deviation from that value, $\Delta CtR_i = \frac{1}{n} - CtR_i$, could be corrected through the expression in Eq. (36) (if $a \neq 0$) or Eq. (37) (if $a = 0$). The following algorithm computes the $\{CtR_i\}$ vector, determines for which leg i the deviation $|\Delta CtR_i|$ is greatest, and computes the corresponding $\Delta\omega_i$ that reduces such deviation in the next iteration. The algorithm stops when either the desired accuracy has been achieved, or iterations exceed a user-designated limit. Figures 11 and 12 show how weights converge to their optimal values for the examples of ERC hedging and trading baskets discussed in Section 3 and 4.

```python
#!/usr/bin/env python
# ERC basket optimization algo
# On 20120516 by MLdP <lopezdeprado@lbl.gov>

import numpy as np
import scipy as sp


#-----------------------------------------
#-----------------------------------------
class ERC:
    def __init__(self,covar):
        # Class constructor
        self.covar=covar
        self.weights,self.ctr,self.ctb,self.tErr,self.iter=0,0,0,0,0
#-----------------------------------------
    def solve(self,precision,seed):
        # Compute the ERC weights and stats
        w=np.zeros(shape=(seed.shape[0],seed.shape[1]))
        w[:]=seed[:]
        iter,n,grad=0,len(w),range(3)
        while True:
            #1) Normalize weights
            w=w/w[0]
            #2) Compute CtB, CtR
            iter+=1
            risk=float((w.transpose()*self.covar*w)[0,0])**0.5
            tErr,mErr,iErr=0,0,0
            ctb,ctr,err=[0 for i in range(n)],[0 for i in range(n)],[0 for i in range(n)]
            for i in range(n):
                for j in range(n):
                    ctb[i]+=w[j,0]*self.covar[i,j]/float(self.covar[i,i])**.5
                ctb[i]=ctb[i]/risk*get_Sign(w[i,0])
                ctr[i]=abs(w[i,0])*self.covar[i,i]**.5/risk*ctb[i]
                err[i]=1./n-ctr[i]
                tErr+=err[i]**2
                # Determine which leg to change
                if abs(err[i])>abs(mErr):
                    mErr=err[i]
                    iErr=i
            #3) Exit conditions
            tErr=(tErr/n)**.5 # Root mean squared deviation
            if tErr<10**-precision:break
            if iter>10**precision:break
            #4) CtR's Taylor expansion
            grad[0]=-err[iErr]
            grad[1]=w[iErr,0]*self.covar[iErr,iErr]/risk**2*(1-2*ctb[iErr]**2)+ \
                self.covar[iErr,iErr]**.5*ctb[iErr]/risk
            grad[2]=self.covar[iErr,iErr]/risk**2*(1-2*ctb[iErr]**2)- \
                ctb[iErr]*w[iErr,0]*(self.covar[iErr,iErr]**.5/risk)**3*(2-3*ctb[iErr]**2)
            #5) Step size
            if grad[2]!=0:
                delta=(grad[1]**2-4*grad[2]*grad[0])**.5
                if grad[1]>=0:
                    delta=(-grad[1]+delta)/(2.*grad[2])
                else:
                    delta=(-grad[1]-delta)/(2.*grad[2])
```

```
            else:
                delta=-grad[0]/grad[1]
            w[iErr,0]+=delta
        self.weights,self.ctr,self.ctb,self.tErr,self.iter=w,ctr,ctb,tErr,iter
#-----------------------------------------
    def get_Seed(self,col):
        # Seeds for hedging basket, based on alternating OLS weights
        a=np.zeros(shape=(self.covar.shape[0],self.covar.shape[1]))
        a[:]=self.covar[:]
        a=sp.delete(a,col,0)
        a=sp.delete(a,col,1)
        a=np.linalg.inv(a)
        b=self.covar[:,col]
        b=sp.delete(b,col,0)
        c=a*b
        c=np.insert(c,col,-1,axis=0)
        return -c
#-----------------------------------------
#-----------------------------------------
def get_Sign(number):
    # Sign of the weight, needed to compute CtB
    if number==0:return 0
    if number>=0:return 1
    if number<=0:return -1
#-----------------------------------------
def main():
    #1) Inputs (covariance, optional seed) --- to be changed by the user
    precision=5
    covar=np.matrix('846960.805351971,515812.899769821,403177.059835136; \
            515812.899769821,351407.396443653,280150.614979364; \
            403177.059835136,280150.614979364,232934.832710412')
    #seed=np.mat(np.ones((covar.shape[0],1))) #--- If a vector of ones is used as seed
    #2) Instantiate class
    erc=ERC(covar)
    #3) Compute and report solution for alternative seeds
    for i in range(covar.shape[0]):
        seed=erc.get_Seed(i) #--- Seed for hedging baskets
        erc.solve(precision,seed)
        # Report results
        print '##### SOLUTION '+str(i+1)+' #####'
        print '##### Weights #####'
        print erc.weights
        print '##### CtR #####'
        print erc.ctr
        print '##### CtB #####'
        print erc.ctb
        print '##### Stats #####'
        print 'RMSD='+str(erc.tErr)
        print '#Iter='+str(erc.iter)
#-----------------------------------------
# Boilerplate
if __name__=='__main__': main()
```

[FIGURE 11 HERE]

20

[FIGURE 12 HERE]

A.3. COMPUTING THE MMSC BASKETS FOR *n*-INSTRUMENTS

What follows is a synthesis of some of the results derived in López de Prado and Leinweber (2012). Please refer to that publication for the detailed proofs.

Let's define the following variables:

$$
\begin{aligned}
a &= \frac{1}{2} \frac{\sigma_i^2}{\sigma_{\Delta B}^2} \left[\rho_{\Delta B, \Delta P_j \omega_j} + \rho_{\Delta B, \Delta P_i \omega_i} \left(2\rho_{i,j} - 3\rho_{\Delta B, \Delta P_j \omega_j} \rho_{\Delta B, \Delta P_i \omega_i} \right) \right] \\
b &= -\frac{\sigma_i}{\sigma_{\Delta B}} \left(\rho_{i,j} - \rho_{\Delta B, \Delta P_j \omega_j} \rho_{\Delta B, \Delta P_i \omega_i} \right) \\
c &= \Delta \rho_{\Delta B, \Delta P_j \omega_j}
\end{aligned}
\tag{38}
$$

Then, for $a \neq 0$:

$$
\Delta \omega_i = \begin{cases} \dfrac{-b + \sqrt{b^2 - 4ac}}{2a} & \text{if } b \geq 0 \\[2mm] \dfrac{-b - \sqrt{b^2 - 4ac}}{2a} & \text{if } b < 0 \end{cases}
\tag{39}
$$

And for $a = 0$:

$$
\Delta \omega_i = -\frac{c}{b} = \Delta \rho_{\Delta B, \Delta P_i \omega_i} \frac{\sigma_{\Delta B}}{\sigma_i \left(\rho_{i,j} - \rho_{\Delta B, \Delta P_j \omega_j} \rho_{\Delta B, \Delta P_i \omega_i} \right)}
\tag{40}
$$

This result allows us to compute the step size $\Delta \omega_i$ for subset i that will change the correlation between subset j and the basket, $\Delta \rho_{\Delta B, \Delta P_j \omega_j}$. This change in subset i can be backpropagated to the legs involved in that subset by multiplying the respective legs' weights by $(1 + \Delta \omega_i)$. By doing so, we can balance the exposure to a subset j, even if it is composed of instruments that are not tradable or subject to constraints.

A.4. THE MMSC ALGORITHM

The hedging basket is determined by minimizing the maximum $\rho_{\Delta B, \Delta P_i \omega_i}$, where $i=1,...,N$ represents any of the N subset of legs (including the n legs). In each iteration, we identify the subset j for which $\Delta \rho_{\Delta B, \Delta P_j \omega_j} = \frac{1}{N} \left(\sum_{i=1}^N \rho_{\Delta B, \Delta P_i \omega_i} \right) - \rho_{\Delta B, \Delta P_j \omega_j}$ is lowest. This is the subset which correlation we would like to bring down to the average value. This can be done by changing the weight of that subset ω_j or, if it is not tradable or its weight is constrained, by changing any other subset's weight ω_i. By reducing our exposure to the subsets with correlations above average, we bring that average down until there is no possibility to keep reducing

$\rho_{\Delta B, \Delta P_j \omega_j}$ without producing some other $\rho_{\Delta B, \Delta P_i \omega_i} > \rho_{\Delta B, \Delta P_j \omega_j}$, at which point the solution has been found.

For a trading basket, the algorithm is essentially the same, with the only difference that at each iteration we identify the subset j for which $\Delta \rho_{\Delta B, \Delta P_j \omega_j} = \frac{1}{N} \left(\sum_{i=1}^{N} \rho_{\Delta B, \Delta P_i \omega_i} \right) - \rho_{\Delta B, \Delta P_j \omega_j}$ is highest. This Maximizes the Minimum Subset Correlation (the purpose of a trading basket), as opposed to Minimize the Maximum Subset Correlation (the goal of a hedging basket).

```python
#!/usr/bin/env python
# MMSC basket optimization algo
# On 20120516 by MLdP <lopezdeprado@lbl.gov>

import numpy as np
from scipy import delete
from itertools import combinations
from math import log
#-----------------------------------------
#-----------------------------------------
class MMSC:
    def __init__(self,covar,maxSubsetSize):
        # Class constructor
        self.covar=covar
        if maxSubsetSize>=covar.shape[0]:maxSubsetSize=covar.shape[0]-1
        self.subsets=self.get_Subsets(range(covar.shape[0]),maxSubsetSize)
        self.weights,self.tErr,self.iter=0,0,0
        self.scorrel=0,0
#-----------------------------------------
    def solve(self,precision,seed,hedge=True):
        # Compute the MMSC weights and stats
        w=np.zeros(shape=(seed.shape[0],seed.shape[1]))
        weights=np.zeros(shape=(seed.shape[0],seed.shape[1]))
        w[:]=seed[:]
        iterTotal,iterW,aim,msc,n,N,grad=0,0,0,1,len(w)-1,len(self.subsets)-1,range(3)
        if hedge==False:msc=0 # For a trading basket
        loop=True
        #1) Iterations
        while loop==True:
            iterTotal+=1
            iterW+=1
            #2) Aim size
            if iterW==1:
                aim=max(1,aim/(1+log(n)))
            else:
                aim=min(10**6,aim*(1+log(n)))
                w[:]=weights[:]
            #3) Iterate subsets
            for i in range(N):
                #4) Normalize weights
                w=w/w[0]
                #5) Compute subset correl matrix
                scovar=self.get_SubsetCovar(w)
                scorrel=get_Correl(scovar)
```

```python
        #6) Determine which leg to change
        avg=(sum(scorrel[:,-1])-1)/N
        err=[avg-scorrel[j,-1] for j in range(N)]
        iErr,tErr=0,0
        for j in range(N):
            tErr+=err[j]**2
            if hedge==True and scorrel[j,-1]>scorrel[iErr,-1]:iErr=j
            if hedge==False and scorrel[j,-1]<scorrel[iErr,-1]:iErr=j
        #7) Store new optimum
        if (hedge==True and scorrel[iErr,-1]<msc) or \
            (hedge==False and scorrel[iErr,-1]>msc):
            weights[:]=w[:]
            iterW=0
            msc=scorrel[iErr,-1]
        #8) CtB's Taylor expansion
        grad[0]=err[iErr]/aim
        grad[1]=-(scovar[i,i]/scovar[-1,-1])**.5*(scorrel[i,iErr]-scorrel[iErr,-1]* \
            scorrel[i,-1])
        grad[2]=.5*scovar[i,i]/scovar[-1,-1]*(scorrel[iErr,-1]+scorrel[i,-1]* \
            (2*scorrel[i,iErr]-3*scorrel[iErr,-1]*scorrel[i,-1]))
        #9) Step size
        if grad[2]!=0:
            delta=(grad[1]**2-4*grad[2]*grad[0])**.5
            if grad[1]>=0:
                delta=(-grad[1]+delta)/(2.*grad[2])
            else:
                delta=(-grad[1]-delta)/(2.*grad[2])
        elif grad[1]!=0:
            delta=-grad[0]/grad[1]
        #10) Backpropagate subset step back to legs
        w=self.get_Backpropagate(w,delta,i)
        #11) Exit conditions
        tErr=(tErr/N)**.5 # Root mean squared deviation
        #if tErr<10**-precision:loop=False # –If exit by convergence
        if iterTotal>10**precision:loop=False
    scovar=self.get_SubsetCovar(weights)
    scorrel=get_Correl(scovar)
    self.weights,self.tErr,self.iter,self.scorrel=weights,tErr,iterTotal,scorrel
#-----------------------------------------
    def get_Seed(self,col,hedge):
        # Seed for basket
        if hedge==True:
            # Seeds for hedging basket, based on alternating OLS weights
            a=np.zeros(shape=(self.covar.shape[0],self.covar.shape[1]))
            a[:]=self.covar[:]
            a=delete(a,col,0)
            a=delete(a,col,1)
            a=np.linalg.inv(a)
            b=self.covar[:,col]
            b=delete(b,col,0)
            c=a*b
            c=np.insert(c,col,-1,axis=0)
            return -c
        else:
            # Seeds for trading basket, based on the leg's betas to the basket
            a=np.zeros(shape=(self.covar.shape[0],1))
```

```python
        for b in range(self.covar.shape[0]):
            a[b,0]=self.covar[b,b]/self.covar[0,0]
        return a
#---------------------------------------
    def get_Backpropagate(self,w,delta,subset):
        # Backpropagates subset delta to the legs
        for leg in self.subsets[subset]:
            w[leg]=w[leg]*(1+delta)
        return w
#---------------------------------------
    def get_Subsets(self,iterable,maxSubsetSize):
        # Generate all subsets up to maxSubsetSize, but including the full set
        subsets=[item for item in combinations(iterable,1)]
        for subsetSize in range(2,maxSubsetSize+1):
            for item in combinations(iterable,subsetSize):
                subsets.append(item)
        for item in combinations(iterable,self.covar.shape[0]):
            subsets.append(item)
        return subsets
#---------------------------------------
    def get_SubsetCovar(self,w):
        # Computes covariances among all subsets
        subsetCovar=np.zeros(shape=(len(self.subsets),len(self.subsets)))
        for i in range(len(self.subsets)):
            for j in range(i,len(self.subsets)):
                for k in self.subsets[i]:
                    for l in self.subsets[j]:
                        subsetCovar[i,j]+=w[k]*w[l]*self.covar[k,l]
                subsetCovar[j,i]=subsetCovar[i,j]
        return subsetCovar
#---------------------------------------
#---------------------------------------
def get_Sign(number):
    # Sign of the weight, needed to compute CtB
    if number==0:return 0
    if number>=0:return 1
    if number<=0:return -1
#---------------------------------------
def get_Correl(covar):
    # Correl matrix from covar
    correl=np.zeros(shape=(covar.shape[0],covar.shape[1]))
    for i in range(covar.shape[0]):
        for j in range(i,covar.shape[1]):
            correl[i,j]=covar[i,j]/(covar[i,i]*covar[j,j])**.5
            correl[j,i]=correl[i,j]
    return correl
#---------------------------------------
def main():
    #1) Input parameters --- to be changed by the user
    precision=3
    covar=np.matrix('846960.805351971,515812.899769821,403177.059835136; \
            515812.899769821,351407.396443653,280150.614979364; \
            403177.059835136,280150.614979364,232934.832710412')
    maxSubsetSize=covar.shape[0]-1 # Set maxSubsetSize=1 for MDR
    hedge=True # True for a hedging basket, False for trading basket
    #2) Instantiate class
```

```
        mmsc=MMSC(covar,maxSubsetSize)
        #3) Compute and report solution for alternative seeds
        for i in range(covar.shape[0]):
            seed=mmsc.get_Seed(i,hedge) #--- Seed for baskets
            mmsc.solve(precision,seed,hedge)
            # Report results
            print '##### SOLUTION '+str(i+1)+' #####'
            print '##### Weights #####'
            print mmsc.weights
            print '##### Subset Correl #####'
            print mmsc.scorrel
            print '##### Stats #####'
            print 'RMSD='+str(mmsc.tErr)
            print '#Iter='+str(mmsc.iter)
            if hedge==False:return
#----------------------------------------
# Boilerplate
if __name__=='__main__': main()
```

Figures 13 and 14 show how weights converge to their optimal values for the examples of MMSC hedging and trading baskets discussed in Section 3 and 4.

[FIGURE 13 HERE]

[FIGURE 14 HERE]

This algorithm can also be used to compute the MDR solution, by setting the parameter *maxSubsetSize=1*. This is equivalent to a MMSC optimization where subsets of more than one leg are ignored. *maxSubsetSize* can also be used to skip evaluating correlations for subsets of larger size, which is convenient should *N* reach an impracticable order of magnitude.

A.5. COVARIANCE CLUSTERING

The number of subsets follows a power law on the number n of instruments involved: $N = \sum_{i=1}^{n-1} \binom{n}{i} = 2(2^{n-1} - 1)$, where we exclude the empty set and the full set. For a sufficiently large n, the number of subsets N to be evaluated per iteration makes MMSC impracticable. But because n large, it also becomes more likely that some of the instruments involved are highly correlated. An approach commonly used to reduce the dimension of a problem applies PCA to identify which orthogonal directions add least variance, so that they can be dropped. There are two arguments to discard such procedure in the context of balanced baskets: First, a key reason for favoring balanced baskets was precisely that they did not require a change of basis, so that the solution could be intuitively connected to the original instruments and subsets of them. Second, "dropping" dimensions involves the loss of information, even if minimal.

In this section we propose a new method for reducing the dimension of a covariance matrix without requiring a change of basis or dropping dimensions. The intuition is to identify which tuples of matrix columns point in neighboring directions, in which case they are redundant and can be clustered together. This can be evaluated by carrying out an eigen-decomposition of the

25

matrix and evaluating which columns have the largest loadings in the orthogonal directions that contribute least variance. When such tuples are clustered together they form a new column that is less redundant, thus contributing to a more parsimonious distribution of the variance across the orthogonal directions. The actual clustering is done by recursively aggregating any two columns (i,j), applying the property that

$$
\begin{aligned}
\sigma[i+j,k] &= E[(i+j-E[i+j])(k-E[k])] \\
&= E[(i-E[i]+j-E[j])(k-E[k])] \\
&= E[(i-E[i])(k-E[k])+(j-E[j])(k-E[k])] \\
&= \sigma_{i,k} + \sigma_{j,k}
\end{aligned}
\tag{41}
$$

$$
\sigma[i+j,i+j] = \sigma_{i,i+j} + \sigma_{j,i+j} = \sigma_{i,i} + \sigma_{j,j} + 2\sigma_{i,j}
\tag{42}
$$

The outcome is a covariance matrix which not only has a smaller dimension but also a smaller condition number, and consequently it is less singular. This is a crucial property, because financial applications typically require the inversion of the covariance matrix, and near-singular covariance matrices are a major source of numerically unstable results.

As explained earlier, this algorithm identifies what column aggregation minimizes the matrix's condition number at each iteration. In this way, we reduce n and, more importantly, N, while making the covariance matrix less singular. One possible procedure would consist in computing, by brute force, all possible clustering outcomes, and determine the one for which the condition number is minimal. That would require $\sum_{i=1}^{m<n} \binom{n}{i}$ covariance clustering operations, where m is the number of dimensions reduced. This is a very large number, and the reason for clustering was to avoid having to evaluate N subset correlations per iteration, so applying brute force does not seem to alleviate our computational problem. An alternative clustering strategy would consist in sequentially pairing columns of the covariance matrix, so that at each iteration we minimize the condition number. That strategy only requires $\sum_{i=1}^{m<n} \binom{n-i+1}{2} \ll \sum_{i=1}^{m<n} \binom{n}{i}$ covariance clustering operations.

More precisely, suppose a covariance matrix V with elements $\sigma_{i,j}$ and $i,j=1,...,n$. We denote the matrix's i-th eigenvalue as λ_i, and its condition number as $c = \frac{\max_i\{\lambda_i\}}{\min_i\{\lambda_i\}} = \frac{\lambda_1}{\lambda_N}$. We would like to reduce V's order to a more manageable $n^* = n - m$, where $2 \le n^* < n$. The following algorithm clusters m elements of V until such requirement is met:

1. If $n = n^*$, return V and exit.
2. For each pair (i,j), $i < j$ of columns of V,
    a. Let $\tilde{V} = V$ be a copy of V.
    b. Insert in $\tilde{V}$
        i. column and row, $n+1$, with k elements, $\sigma_{k,n+1} = \sigma_{k,i} + \sigma_{k,j}$, $\forall k$.
        ii. diagonal element $\sigma_{n+1,n+1} = \sigma_{i,n+1} + \sigma_{j,n+1} = \sigma_{i,i} + \sigma_{j,j} + 2\sigma_{i,j}$.
    c. Strike down columns i,j, giving $\tilde{V}$ an order $\tilde{n} = n - 1$.
    d. Fill symmetric elements of the resulting matrix.

e. $\sigma_{n-1,n-1} = d_{i,j}$.

f. Compute $\{\widetilde{\lambda}_i\}$ for the resulting $\widetilde{V}$.

g. Store the value $c_{i,j} = \dfrac{\widetilde{\lambda_1}}{\widetilde{\lambda_N}}$.

3. Determine the $(i^*, j^*) = arg \max\limits_{\substack{i<j \\ 1<j\leq n}} c_{i,j}$

4. Prepare for the next iteration

    a. Replace V with the matrix $\widetilde{V}$ which clustered together elements (i^*, j^*).

    b. Set $n = \tilde{n}$.

5. Loop to 1.

[FIGURE 15 HERE]

A numerical example with illustrate how the algorithm works. Suppose that we are given a covariance matrix of the 88 most liquid futures. Computing a MMSC hedging basket of that dimension seems quite impracticable. Even though no column may be derived as an exact linear combination of the rest, it is very likely that one column spans a vector in a very close angle to another column or a combination of columns. The algorithm above will identify those situations and form clusters of very similar instruments. Figure 15 shows how, beginning with a numerically ill-conditioned covariance matrix of 88 instruments (determinant above of 10^{305}, condition number of 486,546.1293), the covariance matrix is greatly improved by clustering. For example, for $n^* = 20$, the condition number has dropped to a fraction of its original value (677.41739). As a result, the covariance matrix will now be less numerically ill-conditioned, and n can be reduced to a value for which the MMSC solution can be computed.

We believe this algorithm will prove useful in many financial applications beyond basket construction, like in capital allocation and portfolio optimization problems. What follows is an implementation in Python code. The user simply needs to adjust the statement *path='E:\HFT\Covariance.csv'* with the path where a covariance matrix is stored in *csv* format.


```
#!/usr/bin/env python
# Covariance clustering algo
# On 20120516 by MLdP <lopezdeprado@lbl.gov>

import numpy as np
from itertools import combinations
from scipy import delete


#------------------------------------------
#------------------------------------------
class ClusterCov:
    def __init__(self,cov,minSize):
        # Class constructor
        self.cov=cov
        self.minSize=minSize
        self.cn=[]
        self.clusters=[]
#------------------------------------------
    def solve(self,clusterSize=2):
```

```python
        # Determine the optimal clustering of the covariance matrix
        size=self.cov.shape[0]
        bundles=[]
        while self.cov.shape[0]>self.minSize:
            cn=[]
            comb=self.comb(range(self.cov.shape[0]),clusterSize)
            for trial in comb:
                # Copy matrix
                cov=self.cov[:]
                # Bundle
                cov=self.bundle(cov,trial)
                # Compute condition number
                evalues=np.linalg.eigvalsh(cov)
                cn_=max(evalues)/min(evalues)
                cn.append(cn_)
            # Determine optimal cluster
            sol=self.findSolution(cn)
            if sol<=0:break
            # Store solution
            self.cn.append(sol)
            bundles.append(comb[cn.index(self.cn[-1])])
            self.cov=self.bundle(self.cov[:],bundles[-1])
        self.clusters=self.getClusters(bundles,size)
        return
#------------------------------------------
    def bundle(self,cov,indices):
        # Bundle the covariance matrix
        diag,a=0,0
        # Add clustering column and row
        for i in indices:
            a+=cov[:,i]
        # Compute diagonal element
        for i in indices:
            diag+=a[i]
        cov=self.expandCov(cov,a,diag)
        # Remove clustered columns and rows
        for i in range(len(indices)):
            cov=delete(cov,indices[i]-i,0)
            cov=delete(cov,indices[i]-i,1)
        return cov
#------------------------------------------
    def expandCov(self,cov,array,diag):
        # Expand the covariance matrix
        b=array[:]
        b.shape=(b.shape[0],1)
        cov=np.hstack((cov,b))
        b=np.append(array,diag)
        b.shape=(1,b.shape[0])
        cov=np.vstack((cov,b))
        return cov
#------------------------------------------
    def comb(self,iterable,groupSize):
        # Generate all combinations
        comb=[item for item in combinations(iterable,groupSize)]
        return comb
#------------------------------------------
```

28

```python
    def findSolution(self,cn):
        # Find the optimal solution among candidates
        sol=max(cn)
        for i in cn:
            if i<sol and i>0:sol=i
        return sol
#-----------------------------------------
    def getClusters(self,bundles,size):
        # Determine clusters' constituents
        clusters=range(size)
        for i in bundles:
            a=[]
            for j in i:
                a.append(clusters[j])
            clusters.append(a)
            for j in range(len(i)):
                clusters.pop(i[j]-j)
        return clusters
#-----------------------------------------
#-----------------------------------------
def main():
    # Example using the ClusterCov class
    path='E:\HFT\Covariance.csv'
    #0) Parameters
    minSize=2
    #1) Load the covariance matrix
    cov=np.genfromtxt(path,delimiter=',') # numpy array
    #2) Cluster
    cluster=ClusterCov(cov,minSize)
    cluster.solve()
    3#) Report results
    print cluster.cov.shape
    print cluster.cov
    print cluster.cn
    print cluster.clusters
    return
#-----------------------------------------
# Boilerplate
if __name__=='__main__': main()
```

FIGURES

	FA1 Index	ES1 Index			FA1 Index	ES1 Index
FA1 Index	846960.8	515812.9	FA1 Index		1.000000	0.945486
ES1 Index	515812.9	351407.4	ES1 Index		0.945486	1.000000

*Figure 1 – Covariance and Correlation matrices for
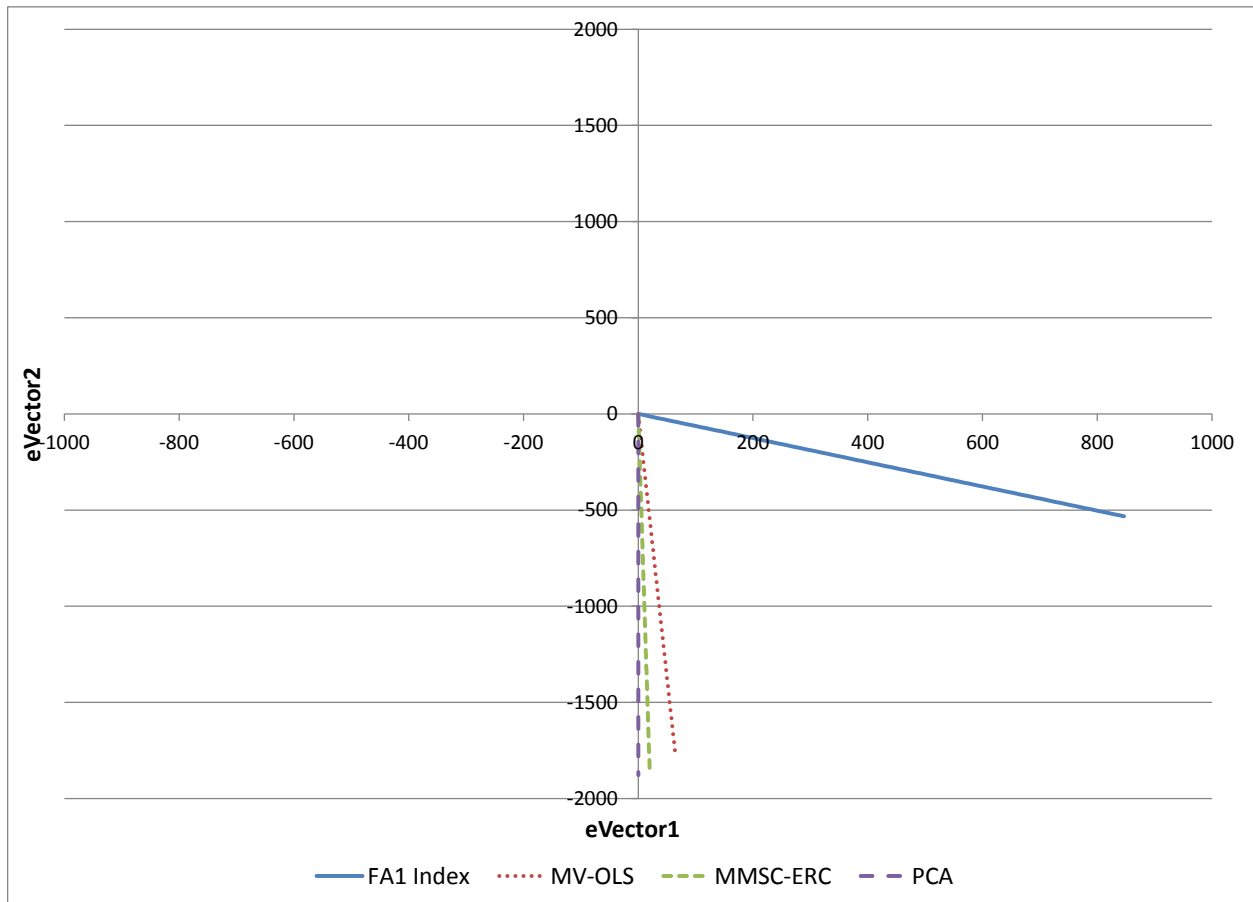the proposed hedging problem (two instruments)*

Figure 2 – Graphical interpretation of the alternative hedging baskets

We can plot the original position (h_1) as a vector in a two-dimensional space characterized by the eigen-decomposition of the covariance matrix from Figure 1. This vector lays in a direction very close to the first eigenvector, which is associated with the market risk component. If we hedge "FA1 Index" with "ES1 Index" using the MV or OLS procedures, the basket's vector spans in a direction very close to the second eigenvector, which is associated with the spread risk. The MMSC and ERC procedures give a solution very similar to PCA's without requiring a basis change.

	FA1 Index	ES1 Index	DM1 Index			FA1 Index	ES1 Index	DM1 Index
FA1 Index	846960.8	515812.9	403177.1		FA1 Index	1.000000	0.945486	0.907710
ES1 Index	515812.9	351407.4	280150.6		ES1 Index	0.945486	1.000000	0.979194
DM1 Index	403177.1	280150.6	232934.8		DM1 Index	0.907710	0.979194	1.000000

Figure 3 – Covariance and Correlation matrices for
the proposed hedging problem (three instruments)

ERC			
INSTRUMENTS	**HOLDINGS**	**CtB**	**CtR**
FA1 Index	1000	0.14	0.33
ES1 Index	-4110	0.05	0.33
DM1 Index	3274	0.08	0.33

Subsets	**1**	**2**	**3**	**1,2**	**1,3**	**2,3**
1,2,3	0.135702	0.051251	0.079026	0.156600	0.102101	0.264190

Figure 4 – One possible ERC solution

The second table in Figure 4 reports the correlation of the basket to each subset of legs, where the subset is identified by the legs is it made of. For example, subset (1,3) is composed of "ES1 Index" and "DM1 Index".

ERC			
INSTRUMENTS	**HOLDINGS**	**CtB**	**CtR**
FA1 Index	1000	0.97	0.33
ES1 Index	1515	0.99	0.33
DM1 Index	1885	0.98	0.33

Subsets	**1**	**2**	**3**	**1,2**	**1,3**	**2,3**
1,2,3	0.969597	0.993448	0.980552	0.995030	0.998350	0.992132

Figure 5 – An alternative ERC solution

DRP					
INSTRUMENTS	HOLDINGS	CtB	CtR		
FA1 Index	1000	-0.49	-0.12		
ES1 Index	18338	-0.62	-1.75		
DM1 Index	-29896	0.77	2.87		

Subsets	1	2	3	1,2	1,3	2,3
1,2,3	-0.492806	-0.619724	0.765364	-0.612224	0.778720	0.983147

Figure 6 – The DRP solution

Although the DRP method is not a balanced basket approach, we report its solution here to illustrate the point that procedures which require a change of basis may yield rather unintuitive solutions.

MDR					
INSTRUMENTS	HOLDINGS	CtB	CtR		
FA1 Index	1000	0.08	0.16		
ES1 Index	-4736	0.08	0.49		
DM1 Index	4031	0.08	0.34		

Subsets	1	2	3	1,2	1,3	2,3
1,2,3	0.075176	0.075176	0.075176	0.142953	0.076735	0.362830

Figure 7 – The MDR solution

MMSC					
INSTRUMENTS	HOLDINGS	CtB	CtR		
FA1 Index	1000	0.19	0.50		
ES1 Index	-3632	0.03	0.22		
DM1 Index	2690	0.07	0.28		

Subsets	1	2	3	1,2	1,3	2,3
1,2,3	0.185136	0.034164	0.074562	0.185136	0.123218	0.185136

Figure 8 – The MMSC solution

F1	ERC	MDR	MMSC	Min Exposure
1	774.88384	774.88384	774.88384	774.88384
2	-2035.20002	-2344.85515	-1798.47923	-1798.47923
3	1286.55426	1584.11666	1057.18793	1057.18793
1,2	-1260.31618	-1569.97131	-1023.59540	-1023.59540
1,3	2061.43809	2359.00050	1832.07176	1832.07176
2,3	-748.64576	-760.73848	-741.29131	-741.29131
1,2,3	26.23808	14.14535	33.59253	14.14535
F2	**ERC**	**MDR**	**MMSC**	**Min Exposure**
1	-620.81023	-620.81023	-620.81023	-620.81023
2	-1969.47783	-2269.13334	-1740.40141	-1740.40141
3	2031.60988	2501.49347	1669.41536	1669.41536
1,2	-2590.28806	-2889.94357	-2361.21164	-2361.21164
1,3	1410.79965	1880.68323	1048.60513	1048.60513
2,3	62.13206	232.36012	-70.98604	62.13206
1,2,3	-558.67818	-388.45011	-691.79627	-388.45011
F3	**ERC**	**MDR**	**MMSC**	**Min Exposure**
1	118.95250	118.95250	118.95250	118.95250
2	2979.10190	3432.37145	2632.59280	2632.59280
3	2222.01391	2735.93534	1825.87424	1825.87424
1,2	3098.05440	3551.32394	2751.54529	2751.54529
1,3	2340.96641	2854.88784	1944.82674	1944.82674
2,3	5201.11581	6168.30679	4458.46704	4458.46704
1,2,3	5320.06831	6287.25928	4577.41954	4577.41954

Figure 9 – Factor loadings for all subsets to the three principal components

MMSC			
INSTRUMENTS	HOLDINGS	CtB	CtR
FA1 Index	1000	0.98	0.50
ES1 Index	0	0.99	0.00
DM1 Index	1907	0.98	0.50

Subsets	1	2	3	1,2	1,3	2,3
1,2,3	0.976655	0.985343	0.976655	0.976655	1.000000	0.976655

Figure 10 – The MMSC trading basket
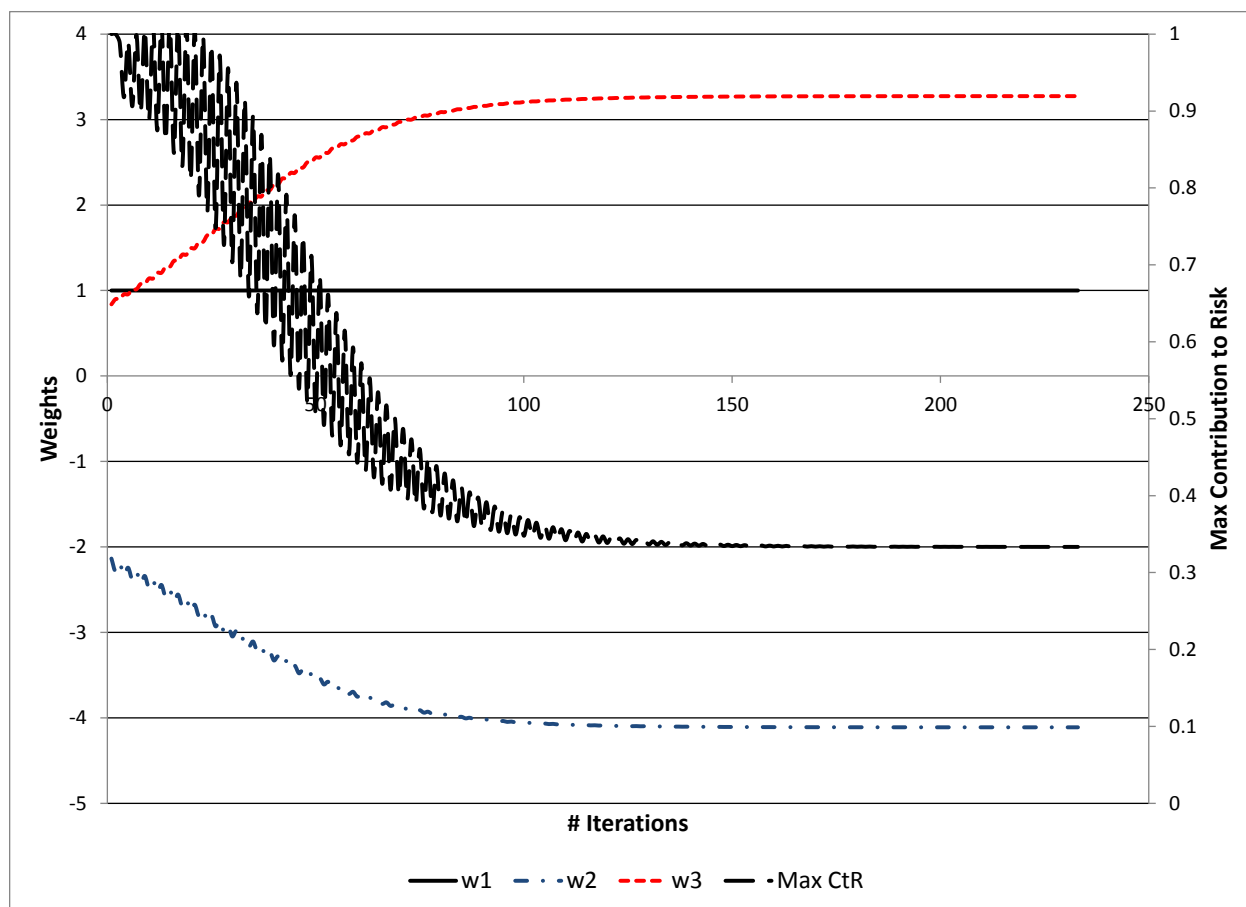
Figure 11 – Convergence of the ERC algorithm when computing the hedging basket

This figure illustrates how the ERC algorithm found the optimal hedging basket after the instruments' weights converged over a number of iterations to a combination such that each leg contributed the same amount of risk.
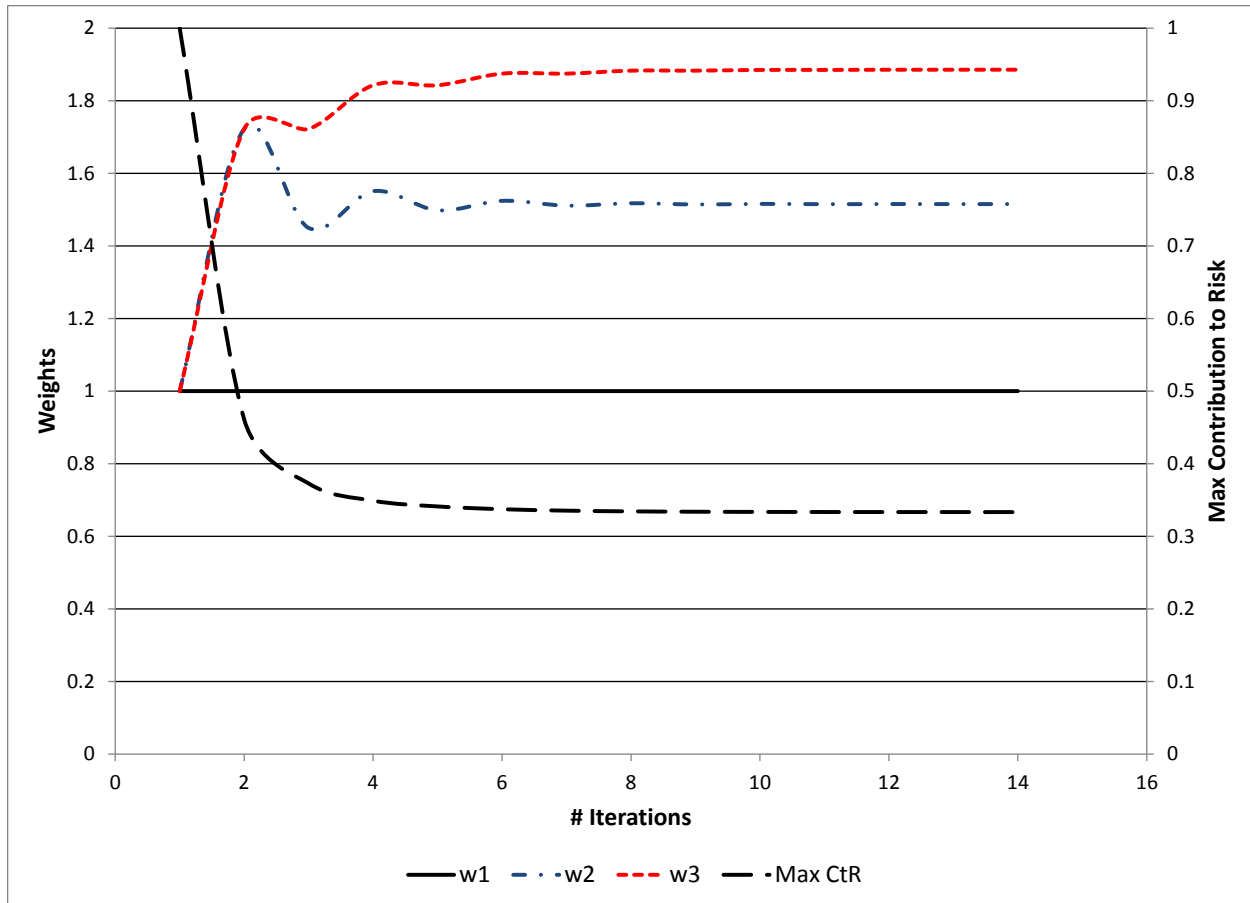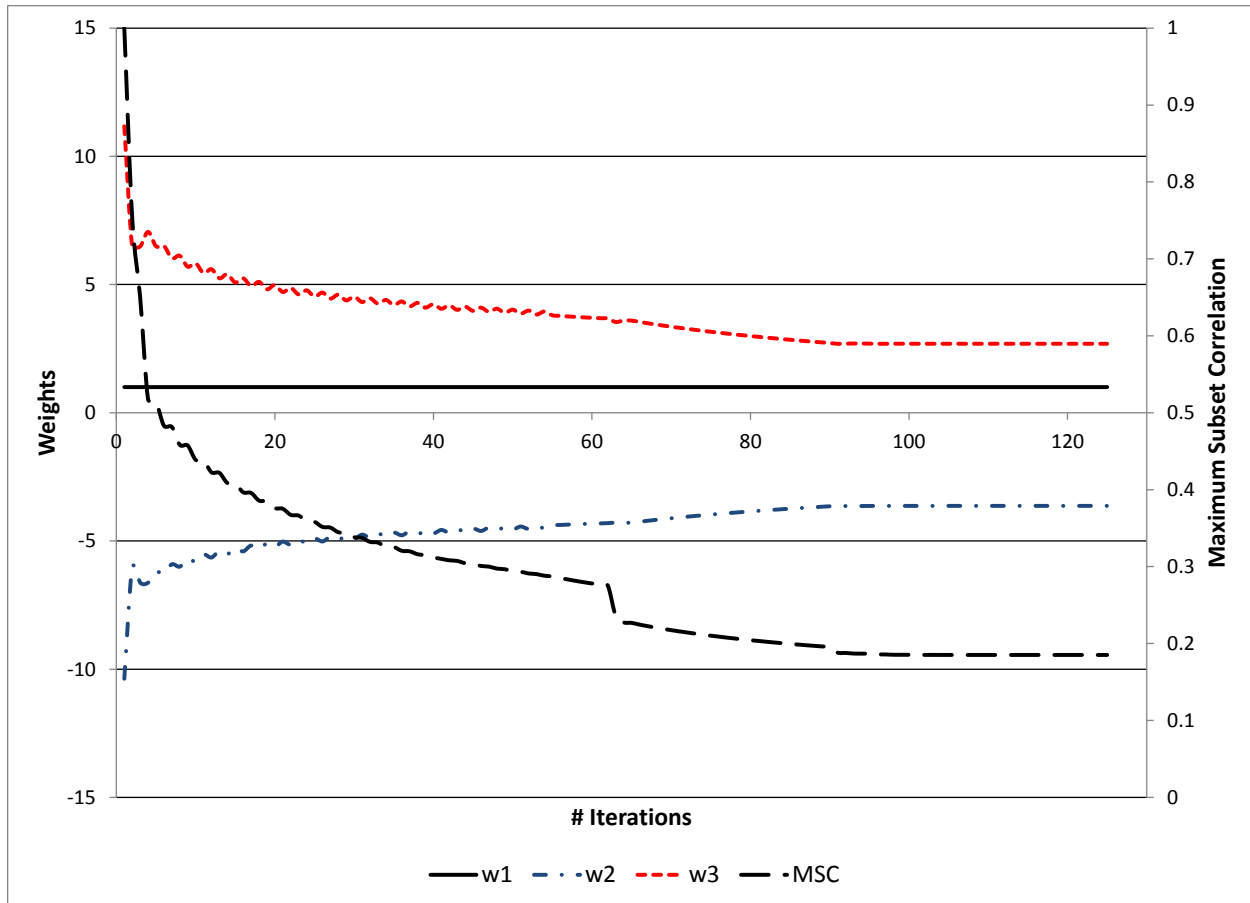
Figure 12 – Convergence of the ERC algorithm when computing the trading basket

This figure illustrates how the ERC algorithm found the optimal trading basket after the instruments' weights converged over a number of iterations to a combination such that each leg contributed the same amount of risk.

Figure 13 – Convergence of the MMSC algorithm when computing the hedging basket

This figure illustrates how the MMSC algorithm found the optimal hedging basket after the instruments' weights converged over a number of iterations to a combination that minimized the maximum subset correlation.
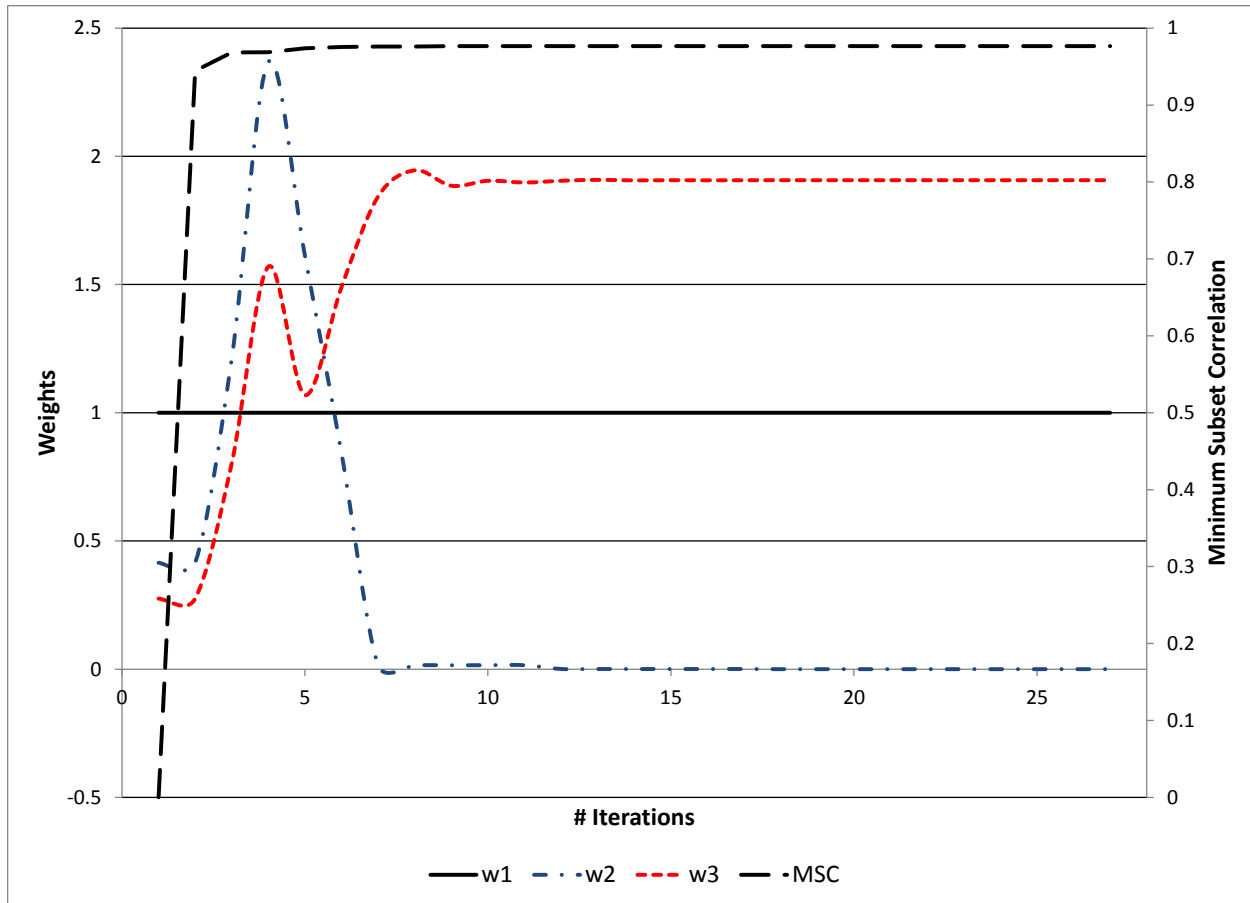
Figure 14 – Convergence of the MMSC algorithm when computing the trading basket

This figure illustrates how the MMSC algorithm found the optimal trading basket after the instruments' weights converged over a number of iterations to a combination that maximized the minimum subset correlation.
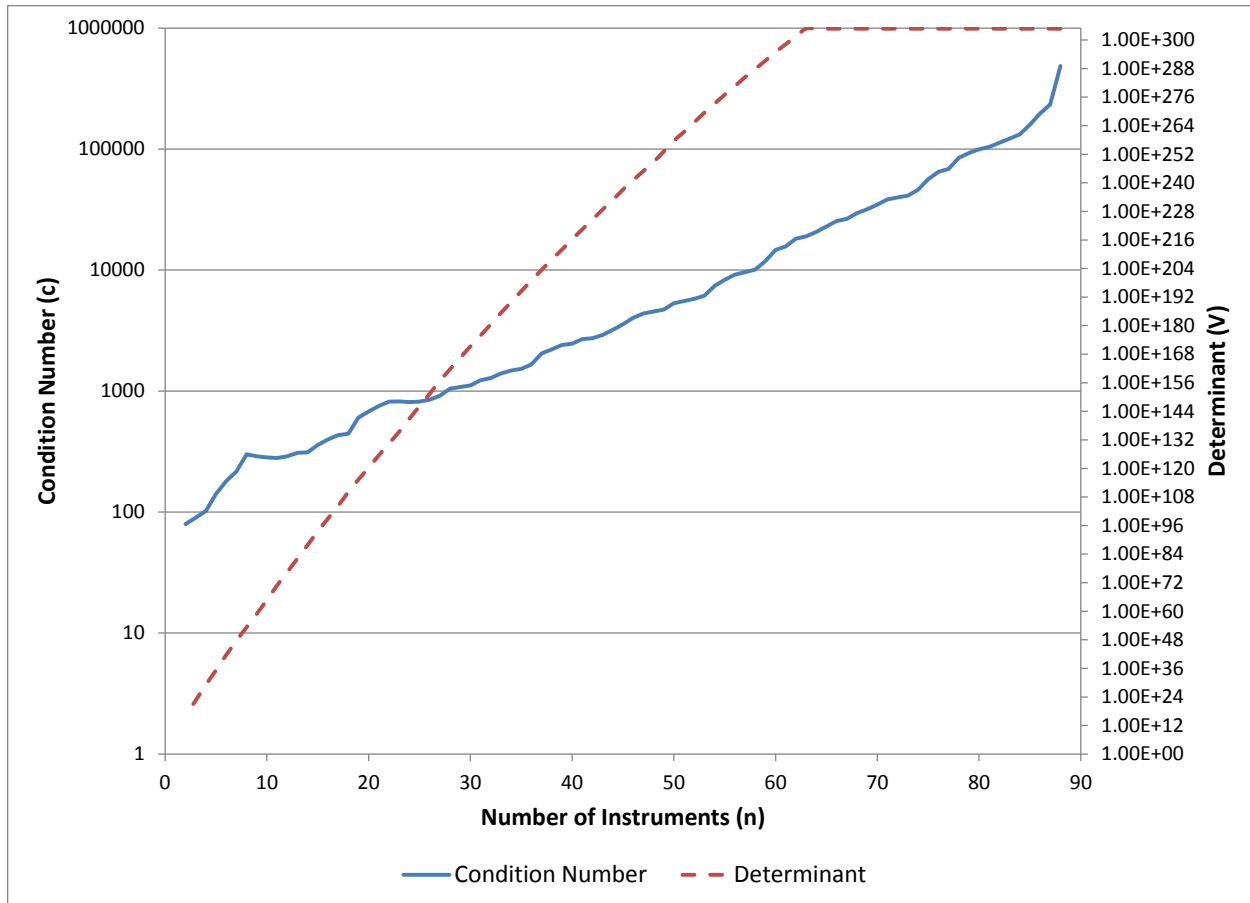
Figure 15 – Reducing the number of instruments (n) by clustering the covariance matrix (V)

This figure illustrates how clustering the covariance matrix improves its numerical condition. We can reduce the number of instruments involved in the MMSC calculation to a value *n* for which the *N* subset correlations can be computed within a reasonable timeframe.

REFERENCES

- Booth, D. and E. Fama (1992): *"Diversification Returns and Asset Contributions"*, Financial Analyst Journal, 48(3), pp. 26-32.
- Clarke, R., H. de Silva and S. Thorley (2011): *"Minimum-Variance Portfolio composition"*, The Journal of Portfolio Management, Winter.
- Demey, P., S. Maillard and T. Roncalli (2010): *"Risk-based indexation"*, working paper. Available in SSRN: http://ssrn.com/abstract=1582998
- DeMiguel, V., L. Garlappi and R. Uppal (2009): *"Optimal versus Naïve Diversification: How inefficient is the 1/N portfolio strategy?"*, Review of Financial Studies, Vol. 22, pp. 1915-1953.
- Choueifaty, Y. and Y. Coignard (2008): *"Toward Maximum Diversification"*, The Journal of Portfolio Management, 34(4), pp. 40-51.
- Choueifaty, Y., T. Froidure and J. Reynier (2011): *"Properties of the Most Diversified Portfolio"*, working paper. Available in SSRN: http://ssrn.com/abstract=1895459
- Hurst, B., B. Johnson and Y. Ooi (2010): *"Understanding Risk Parity"*, working paper. AQR Capital.
- Lohre, H., U. Neugebauer and C. Zimmer (2012a): *"Diversifying Risk Parity"*, working paper. Available in SSRN: http://ssrn.com/abstract=1974446
- Lohre, H., H. Opfer and G. Ország, (2012b): *"Diversified Risk Parity Strategies for Equity Portfolio Selection"*, working paper. Available in SSRN: http://ssrn.com/abstract=2049280
- López de Prado, M. and D. Leinweber (2012): *"Advances in Cointegration and Subset Correlation Hedging Methods"*, Journal of Investment Strategies, Vol.1, No. 2 (Spring), pp. 67-115. Available in SSRN: http://ssrn.com/abstract=1906489
- Maillard, S., T. Roncalli and J. Teiletche (2010): *"On the properties of equally-weighted risk contribution portfolios"*, The Journal of Portfolio Management, Vol. 36, No. 4, pp. 60-70.
- Meucci, A. (2009): *"Managing Diversification"*, Risk, Vol. 22, 74-79.
- Moulton, P. and A. Seydoux (1998): *"Using Principal Components Analysis to Structure Buttlerfly Trades"*, Global Relative Value Research, Deutsche Bank.
- Neukirch, Q. (2008): *"Alternative indexing with the MSCI World Index"*, working paper. Available in SSRN: http://ssrn.com/abstract=1106109
- Scherer, B. (2010): *"A new look at Minimum Variance Investing"*, working paper. Available at SSRN: http://ssrn.com/abstract=1681306.
- Qian, E. (2005): *"Risk parity portfolio: Efficient portfolio through true diversification."* Panagora Asset Management, September.
- Qian, E. (2006): *"On the financial interpretation of risk contributions: Risk budgets do add up"*, Journal of Investment Management, Fall.

DISCLAIMER

The views expressed in this paper are those of the authors and not necessarily reflect those of Tudor Investment Corporation. No investment decision or particular course of action is recommended by this paper.